Ca' Foscari
University
of Venice

Master's Degree Programme

in Language Sciences

(D.M. 270/04)

Final Thesis

Metaphors through vectors: a study on the machine interpretation of (visual) metaphors

Supervisor

Prof. Lebani Gianluca

Assistant supervisor

Dr. Torcinovich Alessandro

Graduand

Francesca Dall'Igna

Matriculation Number 870624

Academic Year

2021/ 2022

INSCRIPTION

Metaphor lives a secret life all around us. […] Metaphorical thinking is essential to how we understand ourselves and others, how we communicate, learn, discover, and invent. But metaphor is a way of thought before it is a way with words.

(Geary, 2012)

To myself

ABSTRACT

Le metafore sono un meccanismo pervasivo che gli umani utilizzano nel loro linguaggio di tutti i giorni. È risaputo come le metafore rappresentino un *mapping* tra due concetti e come sia imperativo trovare un modo efficace per rappresentarle in un ambiente computazionale. Questo studio mira a esplorare una possibile rappresentazione per incrementare l'accuratezza delle macchine nell'interpretazione delle metafore (visive). In modo più specifico, investiga l'uso di vettori per ottenere aggettivi che potrebbero potenzialmente descrivere una data metafora. Inoltre, poiché nel mondo della pubblicità e del marketing l'uso di metafore visive è piuttosto estensivo, questo studio affronta un compito di classificazione di immagini per creare successivamente una *pipeline* da utilizzare nell'interpretazione di metafore visive.

Il dataset di metafore usato per questa tesi è stato ricavato dalla tesi di laurea magistrale di Alice Coli (2016). Le metafore erano rappresentate da settantaquattro coppie *source-target*: alcune di queste erano metafore ben note e utilizzate di frequente, p.e. TEMPO È DENARO; mentre altre erano meno comuni, ma tuttavia altamente comprensibili, p.e. UN ACROBATA È UNA FARFALLA. Come si può capire dalla descrizione del suo progetto, nel dataset raccolto da Coli (2016) sono state ulteriormente analizzate in termini di famigliarità, qualità, innovazione, valenza, concretezza, e comprensibilità.

A tutti questi fattori è stato assegnato un valore da 3 a 7, e il tratto di comprensibilità ha dimostrato come la maggior parte delle metafore fossero altamente comprensibili. Molto probabilmente quando si lavora con le metafore è molto difficile riuscire a raggiungere il valore massimo per quanto riguarda la comprensibilità. Infatti, il meccanismo di *mapping* che sta alla base delle metafore non è un processo diretto e semplice. L'idea di metafora stessa è appunto quella di rompere fino a un certo punto il concetto rappresentato nel dominio *target*.

Sia il lavoro che il dataset di 74 coppie *source-target* erano in italiano, mentre questa tesi e questo lavoro sono stati condotti in inglese. Quindi, una traduzione dei nomi che costituivano le *source* e i *target* è stata portata a termine. I significati delle metafore sono comunque rimasti costanti passando da una lingua all'altra; perciò, non si è sentita la necessità di eliminare alcune coppie dalla lista, in quanto la traduzione aveva mantenuto intatte le relazioni tra dominii.

Oltre al dataset costituito da Coli (2016), sono state aggiunte ulteriori dieci metafore nel dataset finale, a causa di alcune difficoltà nel trovare immagini rappresentati i dominii nella parte di *Computer Vision* del progetto. Queste dieci metafore sono state scaricate dal sito leverageedu.com (Sidrah, 2021). La pagina nel sito di riferimento contiene le metafore più

comuni in inglese, ordinate dalle più semplici alle più complesse. In particolar modo, questo sito può essere utilizzato nella preparazione per le certificazioni di lingua inglese, visto che le metafore elencate sono spesso usate nella sezione di *Use of English* dei suddetti esami; tuttavia, è utile in genere agli studenti che vogliono migliorare e arricchire le loro conoscenze sulle metafore ed espressioni idiomatiche inglesi.

La teoria sulla rappresentazione delle metafore in un ambiente computazionale è stata testata attraverso due esperimenti, dove il primo ha dato risultati inconcludenti. L'esperimento 1 voleva raggiungere lo scopo finale nel modo più semplice possibile, senza creare nuovi funzioni o metodi computazionali troppo costosi. Per ogni *source* e *target* si è creata una lista, dove tutti gli aggettivi relativi a quella *source* o a quel *target* sono stati inseriti. Gli aggettivi sono stati scaricati da Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004). attraverso l'opzione Word Sketch, mentre il corpus di riferimento era il ukWAC 2007 (Ferraresi, Zanchetta, Bernardini, & Baroni, 2008). Gli aggettivi sono stati successivamente filtrati attraverso un set di Python per tenere solo gli aggettivi descrittivi. In altre parole, nomi che cambiavano classe per diventare aggettivi o aggettivi relazionali (ovvero, aggettivi derivati da nomi) o verbi sono stati esclusi. Questo set è stato creato attraverso NLTK (Loper & Bird, 2002) e WordNet (Miller, Beckwith, & Fellbaum, 1990): all'inizio del codice è stato inserito un set vuoto; poi, un'iterazione su aggettivi ('a') e satelliti ('s') ha condotto ad iterare ulteriormente su tutti i synset di Wordnet che erano classificati con quelle *parts of speech*; successivamente, soltanto i lemma sei suddetti synset sono stati selezionati; infine, se l'aggettivo risultava relazionale – ovvero, se derivava da un nome – veniva bypassato dal codice, altrimenti veniva inserito nel set definito all'inizio. L'ultimo step dell'esperimento 1 è stato quello di comparare le liste di aggettivi dei domini di *source* e *target* in una certa coppia. Il confronto mirava a cercare aggettivi che fossero comuni ad entrambi i dominii, visto che le metafore scelte per il dataset erano tutte convenzionali.

Questo metodo è sicuramente semplice e veloce; tuttavia, non ha dato i risultati sperati. È stato condotto un test su sei coppie *source-target* scelte in modo casuale, ma nessun pattern significante o affidabile è stato rinvenuto. Dunque, è giusto presupporre che il primo esperimento sia da classificare come inconcludente.

L'esperimento 2, invece, ha fatto uso di una funzione creata appositamente su Python, che usa la manipolazione di vettori e la similarità coseno per ritornare alcuni aggettivi che potrebbero (così come no) descrivere una data metafora. Innanzitutto, è stato creato un dizionario con le *source* come chiavi e gli aggettivi associati alla *source* come valori, in modo tale che l'accesso a tali aggettivi risultasse più semplice per la funzione. L'idea alla base dell'esperimento 2 tiene

in considerazione solo gli aggettivi relativi alla *source*, non quelli relativi al *target*: lo scopo sarebbe quello di imitare il processo di *mapping* che avviene nel cervello umano, quando si crea una metafora. Dato il fatto che gli umani trasferiscono alcune qualità (ovvero, aggettivi) della *source* al *target*, l'esperimento 2 cerca di replicare questa attività. La funzione creata riceve in input gli aggettivi di una data *source* e il *target* come stringa, presi da una data coppia di *source-target*. In seguito, la funzione ottiene i vettori per ogni parola da fastText e mette in atto sia una moltiplicazione che un'addizione tra il *target* e ogni aggettivo. In questo modo, è possibile ottenere dei vettori risultati che rappresentano un *target* modificato. Infine, la funziona calcola la similarità coseno tra il *target* 'puro' e ogni *target* 'modificato' per ogni coppia *source-target*, ritornando i dieci aggettivi che avevano portato ai valori più alti di similarità.

Per quanto riguarda la parte di *computer vision*, il compito di classificazione d'immagine è stato raggiunto attraverso l'allenamento di ResNet50 su un dataset personalizzato; le immagini, invece sono state ottenute dal dataset ImageNet (Deng, et al., 2009). Sfortunatamente, visto che molte *source* e *target* sono rappresentati da nomi astratti – il che rende difficile la rappresentazione attraverso un'immagine – le metafore del dataset per la parte di *computer vision* erano venti coppie.

Per esaminare l'accuratezza nello scegliere gli aggettivi corretti per ogni metafora, è stato distribuito un questionario su Qualtrics a persone con un livello di inglese almeno B2. Ai partecipanti sono state date undici scelte: le prime dieci erano gli aggettivi ritornati dalla macchina, mentre l'undicesima era un box vuoto dove loro potevano inserire altri aggettivi, che secondo la loro opinione risultavano consoni a descrivere la metafora. Le risposte sono state analizzate usando *precision at k*. I risultati hanno mostrato come i partecipanti fossero generalmente d'accordo con le scelte della macchina. In generale i partecipanti hanno scelto almeno una delle prime dieci scelte per ogni domanda, provando come la macchina abbia sempre scelto almeno un aggettivo corretto. Sebbene gli individui abbiano inserito talvolta ulteriori aggettivi nel box, per nessuna delle 84 domande hanno scelto come unica risposta l'undicesima. L'analisi di precisione che assegnava valore 1 ad ogni aggettivo che avesse ricevuto almeno 1 voto dai partecipanti ha riportato una *precision-at-10* pari a 61%. Inoltre, un'analisi di regressione multipla ha sottolineato come l'indice di 'innovatezza' della metafora fosse legato alla precisione della funzione: più innovative erano le coppie metaforiche, più precisa era la selezione di aggettivi che potevano descriverle. Questo potrebbe avere dei risvolti positivi sull'interpretazione delle *novel metaphors*. Di conseguenza, in accordo con la decisione dei partecipanti, la teoria per l'interpretazione metaforica della macchina è affidabile.

Per quanto riguarda la parte delle immagini, la rete neurale ResNet50 ha riportato un'accuratezza nella fase di testing dell'82%, in linea con i risultati nel campo del riconoscimento di immagini e notevole data la dimensione ridotta del dataset. Anche le singole classi hanno dimostrato delle buone accuratezze in generale, dimostrando come una *pipeline* possa essere plausibile nell'interpretazione di metafore visive.

I risultati suggeriscono come la funzione veloce dal punto di vista computazionale per l'interpretazione delle metafore funzioni, e come l'approccio possa essere implementato in modelli più complessi nell'ambito della comprensione del linguaggio naturale. Su questa base, il concetto teorico alla base di questo studio dovrebbe essere tenuto in considerazione nel creare modelli per l'interpretazione metaforica.

Table of Contents

List of Figures

List of Tables

1. LITERATURE REVIEW

1.1 The Theory of Metaphors

1.1.1 What is a metaphor?

Describing a metaphor seems to be particularly demanding. Were a person to ask a friend what a metaphor is, the friend would most likely make a famous example coming from a classic novel, or a poem, rather than a song. Plausibly, every English-speaking person knows the metaphor written by William Shakespeare "All the world's a stage, and all the men and women merely players" (Shakespeare, 1623). If they were to explain this metaphor, they would probably say that the author is comparing 'the world' to a stage and that humans on this planet are the actors. The interpretation is technically correct, however, this is not the explanation of what a metaphor is.

The study of metaphor as it is now established, did not begin centuries ago; indeed it started with arguably the greatest psychiatrist of all times: Sigmund Freud (Freud, 1914). In his *Zur Einführung des Narzißmus*, Freud clearly states how the study of metaphors is a necessity for the advancement of psychoanalytic theorising. Especially, when interpreting dreams, Freud maps the meaning of the dream into the waking life, which exactly the same process of linguistic metaphors. Nevertheless, this view on metaphors was naturally, and rather obviously, a standpoint of psychiatry and psychology.

The most known theory on metaphors, linguistically speaking, was made by George Lakoff, an American cognitive linguist and philosopher. In 1980, he published one of his most famous works *Metaphors We Live By*, written with Mark Johnson. This book became revolutionary because of the approach and view it gave to the metaphors, especially with the exploration of "conceptual metaphors".

Generally speaking, in cognitive linguistics, a metaphor is described as "understanding one conceptual domain in terms of another conceptual domain" (Kovecses, et al., 2010). An example may be interpreting the concept of life or love in terms of the concept of journey. Basically, speakers interpret 'life' as if it were a 'journey', a very unnatural task to do, but that for humans seems to work perfectly fine.

The work by Lakoff and Johnson is revolutionary for one simple reason. Metaphors had always been treated to be proper to language exclusively, as if they were a mere linguistic phenomenon. Nevertheless, the authors went against this view, by considering that behind the

linguistic expression itself, there was a mapping between two concepts, where one was interpreted by the means of the other. And this view, which to this day is still long-standing, has become known to the general public as the view of 'conceptual metaphor'.
The first example Lakoff and Johnson (1980) make is ARGUMENT IS WAR[1]. In the everyday language, speakers do not use the original conceptual metaphor as a realisation of the mapping processes between the two concepts, but creates linguistic expressions which are the realisation of the main conceptual metaphor. Examples of lingusitic realisations of the before-mentioned conceptual metaphors can be found below (Lakoff & Johnson, 1980, p. 4).

- *Your claims are indefensible.*
- *He attcked every weak point in my argument.*
- *His criticisms were right on target.*
- *I demolished his argument.*
- *I've never won an argument with him.*
- *You disagree? Okay, shoot!*
- *If you use that strategy, he'll wipe you out.*
- *He shot down all of my arguments.*

What happens in cases where speakers pronounce those sentences, and therefore apply the conceptual metaphor above mentioned, is that they do not only talk about arguing or having a discussion with someone, but they do so as if they were at war or in a battle. They are giving a certain shape to the characteristics of arguing: this is where it is possible to see how pervasive conceptual metaphors are and how they model our culture.
In this case, the two individuals involved in the argument are not simply exchanging ideas or taking turns – as if they were having a conversation: they are *attacking the arguments*, and they may *win* (or *lose*) the argument.

This is the reason why metaphors are technically claimed to be a mapping from one concept to another. Speakers are transferring the qualities of one concept into another, so that the receiving argument will be looked at and perceived under a new light. One of the focal aspects about metaphors and metaphorical concepts is their systematicity, i.e. they seem to follow the same structural and cognitive pattern. This feature is vital when it comes to interpret the

[1] From this point onwards: the metaphors written with upper-case letters are the conceptual metaphors; the metaphors written with lower-case letters and in italics are the linguistic expressions of a conceptual metaphor.

metaphor itself. Let us grasp another example of conceptual metaphor and its realizations: TIME IS MONEY (Lakoff & Johnson, 1980, p. 7-8).

- *You're wasting my time.*
- *That flat tire cost me an hour.*
- *I don't have the time to give you.*
- *I've invested a lot of time in her.*
- *You're running out of time.*
- *Is that worth your while?*
- *He's living on borrowed time.*
- *I lost a lot of time when I got sick.*
- *Thank you for your time.*

The second linguistic realisation above-mentioned, *that flat tire cost me an hour*, is a metaphor because the person saying it is not really paying for the time. It is not possible to pay for the time, nor it is possible to lose time, because time is not a concrete object. Money, on the other hand, can be spent, can be earned, can be lost, and can be used to buy other concrete goods or services. Therefore, to give the idea that it was not in the person's intentions to arrive late or make their friend wait for an hour, this person attributes qualities (of money) to time, which normally does not have because of its nature. There was a problem, which took an hour to be solved. However, the same idea is more vividly represented if time is understood in terms of money.

What remains the same throughout all realisations is the attribution of certain qualities of one domain to another, i.e., the systematicity. Because the mechanism of interpreting the metaphor is always the same, speakers select the most striking quality, or the quality with greater emotional impact, of one domain and map it into the other.

Nowadays, these metaphors, as many others, have entered contemporary English dictionaries; speakers do not even recognise them as such, if at all, and they would most likely describe them as "fixed phrases or sentences": this shows how powerful and pervasive metaphors can be and why they deserve research.

The most striking claim that Lakoff and Johnson (1980) make is that it is not only a matter of language, a matter of words. Metaphors do not exclusively stop at a word level, but they extend to all "thought processes" (Lakoff & Johnson, 1980, p. 6). What is meant by the authors is that metaphors usually follow a pattern. Therefore, the conceptual metaphor ARGUMENT IS WAR will be represented by certain actions that speakers do and do not do when arguing.

The reason for this feature is that speakers have internalised and conceptualised the 'argument' in terms of a battle, essentially. This consequently influences not only the way individuals think or reason about arguing, but also how they behave, since humans' behaviour is based on a system of beliefs. If it is claimed that metaphors reflect a conceptual representation, then it is also implied that metaphorical language mirrors what and how individuals think.

One example comes from Thibodeau and Boroditsky (2011), where they tried to investigate how metaphors may – or may not – influence the way speakers think or perceive the world. In their study, participants had to read a description of a crime, which was described either as a 'beast' or as a 'virus' diffusing in a city. Subsequently, they were asked to come up with a solution about how the city should behave to fight this crime.

The researchers discovered that participants exposed to the crime described as a 'beast' were more likely to offer a solition oriented to increasing forces (e.g., more police officers). In contrast, participants exposed to the crime described as a 'virus', offered as a possible solution studying the issue at its root, and understand why such a crime was brought to light in the first place – namely, what scientists do when they study a virus.

This is the main reason why Lakoff and Johnson introduced the idea of conceptual metaphor and they reasoned in terms of metaphorical concepts: "Metaphors as linguistic expressions are possible precisely because there are metaphors in a person's conceptual system" (Lakoff & Johnson, 1980, p. 6).

1.1.2. The source and the target domains

In the previous chapter, it was discussed how metaphors function as the mapping from one domain to another. In technical terms, a metaphor has a source domain (from which speakers select the qualities) and a target domain (to which speakers assign the previously chosen qualities).

In order to understand which domain is the source or target, let us take the two previous examples: ARGUMENT IS WAR and TIME IS MONEY. In the first case, 'war' is the source domain and 'argument' is the target domain, while in the second case, 'money' is the source domain and 'time' is the target domain. To make it easier and more comprehensible, Lakoff and Johnson (1980) assign to conceptual metaphors a mapping of the type: TARGET-DOMAIN IS SOURCE-DOMAIN. The source domain can be a noun, as in the examples just suggested, or it can also be a preposition (e.g., directional metaphors, which will be discussed in the next section). Usually, source domains tend to be more concrete, they may refer to an object, e.g., 'money', although it is possible to find more abstract concepts, e.g., 'argument'. Instead, target domains tend to be more abstract or anyway less delineated: examples may be 'love' or 'time', albeit it is not impossible to find common concrete nouns, e.g., 'person' or 'snow' (see THE SNOW IS A BLANKET).

An interested work has been carried out on the most common source domains by Alice Deignan (1995), where she offered a complete and systematic survey on the topic. Here, only the main source domains will be mentioned.

The first most common source domain is definitely the human body, and usually various parts of it are used as source, e.g., face, legs, head, heart. Linguistc realisations with these sources would be:

- *The heart of the problem*
- *The head of the department*

Another widespread source domain is the domain of animals since it is extremely vast. It seems to be quite productive when it comes to metaphors, as well. Animals are often associated to humans, who are instilled with a specific animal's property. Examples of animals mentioned are *lion, tiger, brute, fox, cow, snake*, etc.

Machines and tools are also often used. Humans in their everyday life use machines and devices very often, smartphones have become an extension of ourselves. The mind or more in general individuals are often associated to a machine (Lakoff, Espenson, & Schwartz, 1991). Examples may be:

- *He had a breakdown.*
- *I wonder what makes him tick.*
- *Fuel up with a good breakfast.*
- *She produces a book every year.*

Other well diffused metaphors involve the domain of heat or of fire as source domains, especially when the metaphor takes emotions and passions as target domains. Rage, love, anger, hate and passion are often described as *burning;* or a person may be *smoldering with anger*.

Deignan (1995) goes on providing other typical source domains, such as health and illness, plants, buildings and construction, games and sport, money, cooking and food, light and darkness, forces.

Interesting is the fact that source domains do not necessarily have to be more or less concrete objects or concepts. Indeed, sources can also be movements and directions, which may sound counter-intuitive to a certain extent, but it actually is not. Examples may be the following:

- *He went crazy*.
- *She solved the problem step by step*.
- *Our economy is galloping ahead*.

These examples are however different from orientational metaphors (chapter 1.1.3): in these cases, the source domain involves a movement or direction which leads to a change of location or state.

Let us consider the first realisation (i.e., *he went crazy*): the man did not actually move and go to a destination called ‘crazy’; the person simply had a mental breakdown or maybe overreacted to a certain situation. However, as it was previously said about TIME IS MONEY, the idea of this man losing his temper and make a scene is better expressed through *he went crazy*, rather than a simple verb like ‘overreact’.

The same reasoning and type of survey which Deignan (1995) carried out for source domains, was implemented for target domains as well, the other half of the metaphor. Generally speaking, it would be possible to claim that target domains are usually abstract, they do not have a clear delineation; therefore, the association to a more defined domain (i.e. the source) allows the target to have more structure and shape.

Among the most diffused target domains we find feelings or emotions, in all their declinations. These types of concepts are primarily understood in terms of another concept, i.e., through a metaphor. An interesting fact about emotions as target domain is that their source domain usually involves movements or forces, as in *he unleashed his anger* or *she was deeply moved*. It is not too farfetch, though, since the etimology of the word itself involves movement. 'Emotion' derives from French 'émotion', which is a derivation of 'émouvoir' meaning 'to set in motion'. The French words derive from Latin 'e-' meaning 'out' and 'movere', which means indeed 'to move' (Treccani, 2021).

Other two distinguished domains which often appear as targets are morality and thought. While morality (often including goodness and badness) is usually understood in terms of concrete concepts, e.g., economic transactions, light or darkness, orientations (example: *I'll pay you back for this*), thoughts are interpreted in terms of less concrete services. The main reason why this is the case is that researchers still do not know much about how thoughts are represented in the brain and how exactly thinking works. Therefore, it is quite natural that speakers try to grasp them by using metaphors: usually, rational thinking is seen as work, while more passive aspects of thoughts are represented through perception, e.g., seeing.

Examples for this target may be:

- *She's grinding out new ideas.*
- *He hammered the point home.*
- *He searched for the memory.*
- *I see your point*

Deignan (1995) continues providing other target domains which appear the most frequently, including: economy, human relationships, communication, time, life (or death) and religion.

Another interesting and fundamental feature of metaphors is the fact that the relationship between targets and sources is not reversible. Considering the conceptual metaphor ARGUMENT IS WAR, it is not possible to find another conceptual metaphor with the same concepts but with reversed roles, e.g. WAR IS ARGUMENT. Indeed, the mapping from source

to target is usually unidirectional, and even if it is possible to revert the roles, either the metaphoricity disappears or the metaphor completely changes the value of its linguistic realisations.

For instance, the metaphor ANGER IS A STORM, has some linguistic realisations, which may be (Kovecses, et al., 2010):

- *It was a stormy meeting.*
- *He stormed out of the room*

It is possible to revert the roles between the target and the source, obtaining A STORM IS ANGER; however, the linguistic realisations will be subject to change as well (Kovecses, et al., 2010):

- *Those are some angry waves.*
- *The storm was raging for hours*

Kovecses and colleagues (2010), nevertheless, suggest that there are certain conceptual metaphors, which are actually reversible. These are generally of the type NOUN IS NOUN, meaning both the source domain and the target domain are represented by nouns in the language. The researchers take as an example the realisation *This surgeon is a butcher*. This expression of the conceptual metaphor is perfectly reversible, by just switching the two nouns, *This butcher is a surgeon*. However, what is possible to remark is that the meaning does not remain intact, it actually shifts. Both sentences are readily accepted, but they are not synonymous.

The reason for this phenomenon given by the authors is that both nouns in the linguistic expression belong to the same "level of abstraction and […] they represent a particular 'meaning focus' in their source domain status" (Kovecses, et al., 2010, p. 28). Let us consider the source and target of the realisation *This surgeon is a butcher*: both 'surgeon' and 'butcher' identify two individuals, with different work positions, who have a role in society. The "level of abstraction" mentioned before (Kovecses, et al., 2010, p. 28) refers to whether both nouns refer to similar entities. Therefore, a metaphor is reversible exclusively when both source and target represent equally abstract – or concrete – nouns: e.g., both indicate humans, both indicate feelings, both indicate objects, etc.

As it was previously mentioned, in order for speakers to understand the mapping between the target domain and the source domain, there must be an interaction between the two sides, otherwise the metaphor would not be understood and the conversation would most likely break down. There is, nevertheless, a rule followed by the mapping between the two domains, called "The Invariance Principle" (Lakoff, 1994). The author describes this principle as the fact that metaphorical mappings "preserve the cognitive topology of the source domain" (Lakoff, 1994, p. 13). This principle is extremely important, almost vital in the comprehension of metaphors, because it gives constraints to the mapping itself, which means it is not as free as speakers may think.

The rule, therefore, guarantees what follows: interiors are mapped onto interiors, exteriors are mapped onto exteriors, and never the other way round. If the metaphor is directional, or if the metaphor involves a path, sources are mapped into sources, goals are mapped into goals, and the list continues. Indeed, it is not possible to find a case where, for instance, the source domain interior will map onto a target domain exterior. Where the Invariance Principle not to be respected, the metaphor would not make any sense. Following the example given by Lakoff (1994), let us consider the metaphor CLASSICAL CATEGORIES ARE CONTAINERS, which always has to respect the classical syllogism of "If X is in category A, and A is in category B, then X is in category B" (Lakoff, 1994, p. 11). If a metaphor describes a relation where B is in X, the syllogism is not respected – just like the Invariance Principle – and the metaphor will not sort its effects. An example would be that of the conceptual metaphor SHAPES ARE CONTAINERS, for which a possible realisation would be *It was a block of chocolate in the form of a cable car*. Note that, however, it is not possible to say *the form of the cable car had a block of chocolate inside*. To put it in Lakoff's words, "This simply does not happen" (Lakoff, 1994, p. 13).

1.1.3 Different types of linguistic metaphors

The metaphor realm is very diversified, the members can be more or less conventional, they may have different functions, and their level of generality can change substantially, i.e., some metaphors are more general than others.

The first big category with which it is possible to classify metaphors is their conventionality. As Kovecses et al. (2010, p. 34) suggest, the conventionality of a metaphor can be described with the "usage [of that metaphor] of a linguistic community". As it can be imagined, both conceptual metaphors and their linguistic realisations are subject to a certain degree of conventionality.

Let us consider some other examples, different from those in the previous two sections: IDEAS ARE FOOD and LIFE IS A JOURNEY. For the first conceptual metaphor, a realisation may be: *I can't digest all these facts*; while for the second conceptual metaphor, the linguistic expression may be of the sort: *He had a head start in life*. It is certainly possible to affirm that these metaphorical expressions are highly conventional, meaning they are well established in the English-speaking community, and speakers may use them quite frequently. As is it was previously assumed in the first section, were speakers to classify these types of sentences, they would most likely classify them as 'collocations' or 'fixed phrases'; it is very unlikely that they would describe them as metaphors. It is obvious to a certain extent that since these expressions are highly conventionalised, they also have an impact on the way speakers think about or understand a certain concept, e.g., ideas or life. Therefore, it would be possible to claim that the higher the conventionality of a certain conceptual metaphor, the greater will be the impact on the understanding of the concepts involved in the said metaphor.

Conventional metaphors stay on one side of the "scale of conventionality" (Kovecses, et al., 2010, p. 35), however the scale has another side, on which it is possible to find unconventional metaphors, known as novel metaphors. Novel metaphors do not necessarily have to arise from novel conceptual metaphors. The conceptual metaphor may be conventional, e.g., LIFE IS JOURNEY, however, the linguistic realisation is not conventional. An example in this case may be (Frost, 1916):

Two roads diverged in a wood, and I –
I took the one less travelled by,
And that has made all the difference.

Now, considering this example, which is a passage of the poem *The Road Not Taken*, by Robert Frost (1916), it is clear that the poet uses the conceptual metaphor of LIFE IS JOURNEY, however, the way it expresses it, the linguistic realisation he employs to do so, is highly unconventionalised. Most likely, before him, nobody had used the same words, in that order, to express the conventional conceptual metaphor mentioned.

That being said, Frost was not the first and will not be the last poet or writer to invent a novel metaphorical expression for a conventional conceptual metaphor. It is, indeed, widely hold that the vast majority of novel metaphors are generated in literature or in general-public-related worlds, e.g., politics (Kovecses, et al., 2010).

What is fascinating, though, is that it is fairly easy to find novel metaphorical expressions for highly conventionalised conceptual metaphor, however, it is extremely hard to find novel conceptual metaphors. Kovecses and colleagues (2010, p. 36) take into consideration the conceptual metaphor LOVE IS A JOURNEY, and while the target domain (i.e., love) can be associated to other source domains to form equally conventionalised metaphors, it is not possible to find other source domains in order to creat novel conceptual metaphors.

Other coventionalised conceptual metaphors involving LOVE as target domain have as sources FIRE (*I'm burning with love*), PHYSICAL FORCES (*He attracts me irresistibly*), NATURAL FORCES (*He was swept off his feet*), ILLNESS (*She has it bad*), MAGIC (*I'm enchanted*), GAME (*He's playing hard to get)*, and so on.

The take-home point in this reasoning is that speakers do not really think of, or understand love in terms of other concepts, aside from those that are conventional. In other words, speakers do not create novel conceptual metaphors with LOVE as target. They may use less conventional source-target pairs, e.g., LOVE IS A COLLABORATIVE WORK OF ART (Lakoff & Johnson, 1980), but they do not create an unconventional conceptual metaphor. The rationale behind it is that speakers understand love perfectly in those terms, and there is no need to look for others. When the experiences are not enough to comprehend a certain target domain, then speakers create a novel conceptual metaphor to better understand it and maybe even explain it.

The second category which is worth mentioned if the intention is that of classifying conceptual metaphors is their cognitive function. Cognitive function is defined as "the function of metaphor for ordinary people in thinking about and seeing the world" (Kovecses, et al., 2010, p. 37). This big category has itself 3 subcategories, in which it is possible to make

metaphors fall in: structural metaphors, ontological metaphors, and orientational metaphors. A brief explanation of the three subcategories will be given, without providing many details.[2]

In structural metaphors, the source domain contains a rich knowledge of the structure for the target domain, or at least relatively. This means, that the (cognitive) function of these types of metaphors is that of making speakers understand the target X in terms of the source Y. This corresponds to the mapping, as it was previously explained in the second section. All the examples provided so far of conceptual metaphors are structural metaphors.

Compared to the previous subcategory, in ontological metaphors the source domain does not provide as much structure knowledge. Their cognitive function is that of giving a "new ontological status" (Kovecses, et al., 2010, p. 38) to the target domain. Speakers assing features proper to objects or substances to many target domains, even though their knowledge about those source domain is rather general or limited. However, speakers find it important to understand many of their experiences in terms of objects or substances. A very simple example can be found in the conceptual metaphor THE MIND IS A MACHINE. Not even researches or neurosurgeons know exactly how our mind appears to be, what its shape is, and so on. Nevertheless, there is a metaphor that assigns the mind to a machine, although it is not clear what type of machine it is talked about.

A more general instance of ontological metaphor is the personification, since speakers assign human features to non-human entitites. Individuals seem to know quite a lot about themselves, and therefore they use this knowledge and apply it to a source domain they know very little of in order to understand it better.

Finally, the last subcategory belongs to the orientational metaphors. These kinds of metaphors provide speakers with even less knowledge about the target domains than the ontological metaphors. These metaphors are called orientational simply because the source domain is a spatial orientation which humans appear to use very often. Examples (Lakoff, Espenson, & Schwartz, 1991) may be:

MORE IS UP / LESS IS DOWN:

- *Speak up.*
- *Keep your voice down.*

[2] For a more detailed description, please refer to Kovecses et al. (2010, pp. 37-42)

HAPPY IS UP / SAD IS DOWN:

- *That boosted my spirit*
- *I fell into depression.*

FORSEEABLE FUTURE IS UP (AND AHEAD)

- *What's coming up this week?*

A *fil rouge* which seems to connect all orientational metaphor is that an upward direction tends to pair with a positive feeling or evaluation, while a downward direction tends to go with a negative evaluation.

The last great category for classifying metaphors, which nevertheless regards conceptual metaphors only, is the generality. As it was previously stated, conceptual metaphors can be either generic-level metaphors or specific-level metaphors. The conceptual metaphors analysed in the previous sections (e.g., LOVE IS A JOURNEY, ARGUMENT IS WAR, IDEAS ARE FOOD) are all specific-level metaphors, because their targets are specific concepts – abstract, but still specific.

Next to these kinds, it is possible to find general-level metaphors. Examples of general-level conceptual metaphors are: EVENTS ARE ACTIONS or GENERIC IS SPECIFIC. They are called generic because the target themselves are general; EVENTS does not mention a specific event, compared to LOVE which was a specific feeling. General targets only have a defined number of qualities. The point is that although these general-level metaphors do not provide as much information as specific-level metaphors do, they still have a remarkable duty: while EVENTS ARE ACTIONS is responsible for many cases of personification, GENERIC IS SPECIFIC accounts for proverbs, because proverbs "often consists of specific-level concepts" (Kovecses, et al., 2010, p. 45). Interesting is that these specific concepts help speakers understand the proverb at a more general level, one that is adaptable to many occasions for instance, which make the proverb itself very well known.

Finally, a very particular type of metaphor, which does not involve language per se, or at least it does so seldomly, is the visual metaphor. The rationale behind how visual metaphors work is not that far from linguistic metaphors, and in any case, in order to understand visual metaphors, speakers need to go through language and verbalise them.

1.1.4 Visual metaphors

A visual metaphor is described as a pictorial analogy (Kogan, Connor, Gross, & Fava, 1980). It provides a comparison between what can be seen on the image and another concept – which may or may not be represented – to express a figurative meaning, as also linguistic metaphors do. Visual metaphors have been extensively used in advertisiment over the past 100 years (Ryoo, Jeon, & Sung, 2021). There are many similarities between linguistic metaphors and visual metaphors. The main common feature is, most likely, that they both have a source domain and a target domain. However, visual metaphors also possess what is called the 'ground'. The source is the object or represented concept in whose terms the target is interepreted. The target is the object or concept that requires the description, while the ground represents to the shared feature between the target and the source (Van Mulken, van Hooft, & Nederstigt, 2014).

To make a clear example, Tide – a famous American brand of detergents – created an advertisement for their product which showed some dipped shirts in a cup containing a liquid blue as the sky. So, in this case, the source is obviously the blue sky, the detergent promoted is the target, while the ground is the freshness and cleanness emphasised and publicised by the advertisers (Ryoo, Jeon, & Sung, 2021).

Just like linguistic metaphors are categorised based on generality, conventionality and function, visual metaphors are classified as well, although there is less consensus on how to do this. Recently, Van Mulken, van Hooft, & Nederstigt (2014), collected all previous studies and suggested three types of visual metaphors: juxtaposition, fusion and replacement. According to the authors, the complexity of the visual metaphors is based on how the source and the target are spatially distributed: in juxtaposition, the source and the target are visually equated; in fusion, the source and the target are combined so that the target is blended within the source; and in replacement, the source is the only component to be represented, i.e., without the target. Unfortunately, visual metaphors are not all prefectly understendable by everyone, some are particularly complex. Since the function of visual metaphors is that of persuading people in buying a certain product or promote a predifined device, if the metaphor itself is not understood, it does not serve its function properly. Therefore, advertisers present the metaphors with some verbal messages, which can be slogans, headlines or a body copy, called 'text anchors'.

An example of text anchor may be the sentence used by Heineken while promoting one of their beers: advertisers created a beer drum thorugh a pile of CDs with a glass of the Heineken beer next to it; however, they also wrote “made to entertain”. The text anchor facilitated the interpretation of the visual metaphor, which would have been otherwise very complex to unfold.

Figure 1. Advertising of Heineken beer with text anchor "made to entertain". The entertainment comes from the pile of CDs representing the can of beer. Retrieved from adsarchive.com

The project which will be described in a few chapters does not involve visual metaphors. However, these types of metaphors are very pervasive, just like linguistic metaphors are. Since the previous chapter was devoted to explaining the different kinds of linguistic metaphors, and since the project does involve images, this brief section inserted itself quite naturally. However, given the fact that the theme of the project is not visual metaphors, their description stops with this chapter.

1.2 Computational experiments with metaphors

With the widespread availability of the theory by Lakoff and Johnson (1980), research in many other interdisciplinary fields tried to grasp in details the outcomes of such an extensive usage of metaphors. Many studies have been conducted on how individuals perceive metaphors (McClintock & Ison, 2004; McGlone, 1996; Boers & Littlemore, 2009) and on what type of consequences there are given the fact that we understand a concept in terms of another (Morris, Sheldon, Ames, & Young, 2007; Erikson & Pinnegar, 2017).

There is, however, a rather recent field which may benefit from further studies on metaphors, naimly Machine Learning (ML), given the fact that devices and computers are becoming extensions of human bodies. In ML, it is possible to distinguish between two main branches: Computer Vision (CV) and Natural Language Processing (NLP) (Torcinovich, A., personal communication, September 13, 2021). In particular, NLP deals with analyses of natural languages from a computer's point of view. The field is itself interesting and has several applications, however, since metaphors are so pervasive, it is fair to say that a great deal of the research should involve metaphors.

The task of making the computer treat in one way or another linguistic metaphors is quite demanding. The reason is not because metaphors are difficult themselves, but because computers do not reason in terms of natural languages. They have a very simple – albeit not easy – language, called binary language, which essentially only has two 'terms': 0 and 1. All the information that computers receive and have to analyse is stored as a combination of these two numbers. Computers do not have the meaning comprehension humans have, at least as far as we know. This itself makes treating metaphors from a computational point of view rather troublesome, especially when the task involves representing metaphors.

Generally speaking, the two main tasks towards which research moves are metaphor recognition and metaphor interpretation. While the first task involves discerning between literal language from metaphorical language in a given expression, the second task consists of determining the intended meaning of a metaphorical expression (Shutova, 2010).

1.2.1 Metaphor identification

One of the first attempts of successful metaphor identification, also known as metaphor recognition or detection, stems from the work of Fass (1991). He created a method called the 'met* method', which recognises "selected examples of metonymy and metaphor, […] in short English sentences" (Fass, 1991, p. 50). Firstly, the method distinguished between literalness and non-literalness by using preference violation. Secondly, if the sentence was detected as non-literal, it was tested for metonymy. Thirdly, if the system did not recognise a metonymy, the sentence was tested for "relevant analogy" (Shutova, 2010, p. 689) in order to find a possible metaphor. Finally, the met* method looked for a triple, containing hypernyms for both the source and the target domains, which would represent a metaphor. There certainly were some limitations to this approach, Fass himself noticed them, more precisely in the preference violation aspect (Fass, 1991). Another relevant problem was to be found in the conventionality of certain metaphors. Because some metaphorical senses are very common in everyday language, the system would "extract selectional preference distributions skewed towards such conventional metaphorical senses" (Shutova, 2010, p. 690). The issue in this case is that some expressions are metaphorical in meaning, however no preference violation can be detected.

Another work worth mentioning is the CorMet project (Mason, 2004). This project was the first attempt to discover an automatical mapping between the source domain and the target domain. The system searched for variations in selectional preferences, drawn from Internet corpora. After the project collected all outputs, Mason compared the results with the Master Metaphor List (Lakoff, Espenson, & Schwartz, 1991) in order to calculate the accuracy of his system. The researcher reported an accuracy of 77%, although subjectivity should be taken in consideration, since the Master Metaphor List is based on hand-made metaphorical mappings between source and target domains.

More recently, different researchers and experts have tried to tackle the issue of metaphor identification from other perspectives, trying to pursue innovativeness at its peak. A work worth mentioning is the one by Neidlein, Wiesenbach, and Markert (2020). The authors conducted a linguistic analysis on a task involving metaphor recognition using systems based on language models. They considered nine models:

- Lex-BL: a baseline based on Gao, et al. (2018);
- Wu (Wu, et al., 2018), based on word2vec (Mikolov, Chen, Corrado, & Dean, 2013);
- Gao (Gao, Choi, Choi, & Zettlemoyer, 2018)

- Mao (Mao, Lin, & Guerin, 2019), which is built upon Gao;
- Dankers (Dankers, Rei, Lewis, & Shutova, 2019), an enhanced version of BERT;
- Stowe (Stowe, Moeller, Michaelis, & Palmer, 2019), based on ELMo (Peters, et al., 2018);
- BERT (Delvin, Chang, Lee, & Toutanova, 2019);
- ILLI (Gong, Gupta, Jain, & Bhat, 2020);
- DM (Su, et al., 2020)

The systems were tested on the VUA Metaphor Corpus (Steen, et al., 2010), where words are annotated as either *metaphorical* or *literal*. The training was conducted on the VUA-ALL-POS set, while the testing on the VUA-SEQ set. The measures for the evaluation were precision, recall and F1. In the traning, overall the systems performed worse compared to the testing phase. On the VUA-ALL-POS set, the best results were achieved by DM-ENS (a modified version of DM) with an overall accuracy of 91.6%. On the VUA-SEQ, the system which performed better was BERT, with an overall accuracy of 94.4%.

The following step was that of conducting an analysis on "how well the current systems handle conventional vs novel metaphors" (Neidlein, Wiesenbach, & Markert, 2020, p. 3726), starting from the standpoint that these systems would have not performed well with non-conventional metaphors. The authors found how the models do not demonstrate generalisation abilities, and perform worse on novel metaphors compared to previous conventional metaphors.

Finally, a compelling excursus on automated metaphor identification has been shared by Leong, et al. (2020), who reported the results of a shared task on metaphor identification on the VUA Metaphor Corpus (Steen, et al., 2010) and on a subset of the TOEFL (Beigman Klebanov, Leong, & Flor, 2018). They adoped three baselines (Leong, et al., 2020, p. 20):

- Baseline 1: UL + WordNet + CCDB
- Baseline 2: bot.zen by Stemle and Onysko (Stemle & Onysko, 2018)
- Baseline 3: BERT (Delvin, Chang, Lee, & Toutanova, 2019)

The report shows how, overall, all teams performed better on the VUA corpus compared to the TOEFL. The authors underline that if TOEFL is considered as an additional genre to the VUA corpus, it is possible to observe that TOEFL's genre is overall more difficult to analyse compared to Academic or News.

The tasks dealing with metaphor identification are arguably the most difficult in the field of NLP, or more generally NLU (i.e., Natural Language Understanding). This is particularly true

for novel metaphors, rather than well established conceptual metaphors. Novel metaphors are not as spread and used as conventional metaphors, which appear frequently and sometimes are not even thought about as metaphors. This is reflected on corpora, which are fundamental when dealing with research on distributions. Novel metaphors are simply not represented in those corpora, because they have never been produced before, or hardly ever at least. Therefore, it is extremely difficult to foresee among the large number of nouns available in a language which are going to be selected as the source and target of a novel metaphor. Indeed, so far, there has not been a setup that allows researchers to tackle the issue as far as novel metaphors are concerned. The main problem lies in the fact that, if a person creates a new metaphor, the said person is creating a mapping between two concepts that normally do not bind together to constitute a metaphor. There have been some steps forward in word-level metaphor identification, nevertheless tasks involving a broader identification of conceptual metaphors have been ignored (Tong, Shutova, & Lewis, 2021).

1.2.2 Metaphor interpretation

Metaphor interpretation has been approached from variously different perspective over time, with some quite successful results. One of the very first was the model suggested by Kintsch (2000). The model unifies three components – latent semantic analysis (LSA), construction-integration (CI) model and "the claim that literal and metaphoric predication are alike" (Kintsch, 2000, p. 265) – into a theory for metaphor interpretation. LSA (Landauer & Dumais, 1997) represents meaning, in a semantic space, by the relationships of one word with other words. Since LSA has its limitations, e.g., the fact that it does not explain the type of relationship between a given word and its neighbours, it has been paired with CI to solve some of the issues. The algorithm suggested in Kintsch (2000) consists in three steps: the first computes the "semantic neighbourhood of P" (Kintsch, 2000, p. 259), with P being a given predicate; the second step constructs an activation network which follows the lines of the CI model, where each neighbour term of P is connected to P, the argument (A) and an inhibitory link. If the activation spreads towards the inhibitory link, most nodes will become deactivated (Kintsch, 2000). Finally, the nodes with the highest activation compute a vector, which will represent the meaning of the metaphor. The most salient result of the psycholinguistic experiment which was conducted in the paper, is the fact that metaphors are "understood directly" (Kintsch, 2000, p. 263); indeed, the model suggested does incorporate this premise. Moreover, the model gives a valid metaphor interpretation-specific model, based on the claim that metaphorical utterances are like literal utterances.

An interesting problem has been raised by Utsumi (2011), where he attempts to answer a straightforward, yet complex, question: "Which property determines the choice between [the categorisation and comparison] processes for metaphor comprehension?" (Utsumi, 2011, p.251). Indeed, research had discovered how metaphor comprehension is based on categorisation and comparison; hence, understanting which views on the topic were the most plausible, was the task proposed by the author. In particular, he considers: the conventionality view (Bowdle & Gentner, 2005), the aptness view (Glucksberg & Haught, 2006) and the interpretative diversity view (Utsumi, 2007). Each view tries to answer the question in a different way, respectively putting the accent on vehicle conventionality, metaphor aptness, and interpretative diversity. Thus, Utsumi (2011) compares the different views using "cogntiive modelling and computer simulation based on a semantic space model" (Utsumi, 2011, p. 251). In the suggested experiment, the two processes were modelled in a semantic space, which is

itself based on the LSA (Landauer & Dumais, 1997). The resulting two models received a vector for the words presented in a metaphor and computed a vector for the meaning of the given metaphor. The final vectors were analysed to investigate the degree to which they resemble human interpretation (of metaphors). Finally, the results of the models were predicted by the three properties highlighted by the three views. The experiment was conducgted in Japanese, and therefore, on Japanese metaphors.

The final results showed that both interpretative diversity (Utsumi, 2007) and vehicle conventionality (Bowdle & Gentner, 2005) affect the choice of one process over the other, while the third view – metaphor apteness (Glucksberg & Haught, 2006) does not seem to affect the choice.

Despite the difficulties and the questions regarding processes in understanding and intepreting metaphors, research has tried to find the perfect way to make machine correctly interpret metaphorical sentences. Two projects have made the papers regarding metaphor interpretation back in the days. In both works, researchers tried to create a reasoning framework based on metaphors following the steps of the theory on conceptual metaphor (Lakoff & Johnson, 1980). The reasoning processes adopted by the two projects were dependent on some knowledge provided by manually-coded information about the world, and mainly engaged the source domain. The results obtained through this process were consequently mapped onto the target domain, as linguistic metaphors naturally do. The first project was the KARMA system (Narayanan, 1999), a then new computational model for verb semantics, which was applied to metaphors as well. According to its creator, the KARMA system was able to use the "metaphorical projections of motion verbs to refere in real-time important features of abstract plans and events" (Narayanan, 1999, p. 1).

The second was the ATT-Meta project (Barden & Lee, 2002), which dealt with metaphorical and metonymic descriptions of mental states or reasoning using first order logic. The interesting characteristic about this system is that it performs some reasonig which is considered necessary to process metaphorical utterances. However, the limitation to this project concerned the fact that the system did not take natural language sentences or phrases as input. The system was fed with logical expressions which could be seen as simple representations of short discourse fragments – which may or may not be metaphorical.

Most recently, metaphor interpretation tasks have been treated creating different and varied tasks. One successful result in this field was reached by Su, Huang, & Chen (2017), who suggested a property-transfering process in the automatisation of both metaphor recognition and metaphor interpretation. The researchers presented a model that deals with nominal

metaphors and recognises whether a given sentence is a metaphor or not: this has a limitation, since the model works only if the source and target have a "the same direct ancestor" (Su, Huang, & Chen, Automatic detection and interpretation of nominal metaphor based on the theory of meaning, 2017, p. 300).

As far as the metaphor interpretation task is concerned, the approach used semantic relatedness between the source and the target. This task has been devided in two steps: extraction and tranfer. The system received a source-target pair and it extracted the source's properties, and then chose a property that was the closest – semantically speaking – to the pair. To make an example, and visualise it, the conceptual metaphor LOVE IS TIDE was interpreted by the computer as *the love is unstoppable* (Su, Huang, & Chen, Automatic detection and interpretation of nominal metaphor based on the theory of meaning, 2017).

Another worth-mentioning example of metaphor interpretation is the work by Bollegala and Shutova (2013). The researchers approached the task by using paraphrases, so that this can be used to replace the word carrying the metaphoricity in a given non-literal sentence. The approach is not a new idea per se, since it had been previously used in supervised training settings using pre-built lexical resources (e.g., WordNet) (Shutova, 2010). In supervised learning it is often possible to achieve high levels of accuracy: in the case just mentioned, the accuracy reached 0.81 (i.e., 81%), which can be considered a good result. However, supervised learning is extremely time-consuming; therefore, Bollegala and Shutova (2013) used the same approach involving metaphor paraphrasing but with unsupervised learning. The method essentially extracts the paraphrases previously created for a given metaphorical expression, which they collect from the Web. The unsupervised technique was suprising because it returned an accuracy of 0.42 (i.e., 42%), which is an extremely good and rather high score in unsupervised settings.

The final approach on metaphor interpretation which is worth mentioning is a recent work by Song et al. (2021). Their study aimed to present a new approach to the field of metaphor processing in NLP using knowledge graph embedding (KG embedding).

KG embedding are meant to embed components of knowledge graphs into continous vector spaces. In this way the structure of the KG remains intact, however the manipulation of its vectors results easier (Wang, Mao, Wang, & Guo, 2017). Song and colleagues start from the point that a metaphor can be seen as a combination of three parts: a source, a target, and an attribute. Therefore, their method first creates a knowledge graph for each metaphor; subsequently they also extract some concept-attribute collocations which will be added to the

knowledge graph of said metaphor, since collocations are important to understand and interpret metaphors (Song, Guo, Fu, Liu, & Liu, 2021).

Once they had created the dataset, they split it with a ratio of 70-10-20. The evaluation consists in masking either the source or the attribute, which in turns needs to be predicted by the mdoel. Finally, this type of embedding was tested on different metaphor-related task: identification, interpretation and generation. Metaphor interpretation and generation tasks involved "completing the [knowledge graphs]" (Song, Guo, Fu, Liu, & Liu, 2021, p. 406), while identification consists in a classification task on "enhanced concept pair[s]" (Song, Guo, Fu, Liu, & Liu, 2021, p. 406). The authors demonstrate how their suggested model enhances the computer performances on metaphor processing, proving the effectiveness of their method.

1.2.3 Consequences for NLP

NLP definitely has to take into consideration that metaphors necessarily require two concepts divided into two different domains. It is therefore essential to this field that both domains are processed automatically when performing tasks involving metaphor, if the ultimate goal is that of making computer recognise or interpret metaphors with the highest percentage of accuracy.

What is more, to the best of my knowledge, there is an area in NLP which has remained quite unexplored, namely metaphor generation. Although it may be correct to say that metaphor generation is encompassed in the more general field of Natural Language Generation (NLG), however a more detailed and specific on the subfield could lead NLG to another level. Because metaphors are extremely pervasive, it is quite obvious that, were researchers to reach an artificial intelligence, this would have to speak just like humans speaks; and this includes metaphors as well. The main issue in this case is the intentionality that states itself behind metaphors: humans use them to express a certain idea. Without a reasoning process or intentionality, making computers generate metaphors is rather complicated. However, with models for metaphor interpretation, metaphor identification and the use of metaphors in human communication, it could be possible to finally find the light at the end of the tunnel.

2. RESOURCEFUL DATABASES AND MODELS

This second chapter serves a very simple purpose. Since many databases, models, and functions have been used to achieve the scope of the thesis, it seemed appropriate to give them the right amount of credit. Therefore, the intention of the following pages is that of introducing the various protagonists of the project, explain their structure and features, as well as describing how and why they were used to reach the aim.

The databases in question are two: WordNet (Miller, Beckwith, & Fellbaum, 1990), which was used both for the NLP and CV sides; and ImageNet (Deng, et al., 2009), which was used exclusively for the CV part.

Moreover, the chapter will introduce fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016), used to retrieve vectors, and Sketch Engine, chosen to retrieve the adjectives.

One important notice has to be made about this chapter: although it is about presenting and describing the roles of said databases and models, it is also going to lay out the reasons for the choice falling on them. The order of introduction will follow the one in which the tools and databases were used during the project – namely, Sketch Engine, WordNet, fastText, and finally, ImageNet.

2.1 Sketch Engine

The first tool used as far as the project is concerned was Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004). Sketch Engine is a text analysis tool: it uses different corpora, of multiple languages, in order to find what can be described as 'typical' in a given language, or in the opposite direction, what is rare and outdated. More specifically, Sketch Engine is a corpus manager and a software for lexical analysis, built by Lexical Computing Limited in 2003. The goal of Sketch Engine is that of allowing linguists and individuals more generally study language behaviour (Kilgrarriff, et al., 2014). The name derives from one of its most used features, i.e., Word Sketch. The software was created by research scientist Adam Kilgrarriff and computer scientist Pavel Rychlý. When Kilgrarriff decided to collaborate with Rychlý, Rychlý was already working in the NLP field, and had already developed Manatee and Bonito. These two from the architecture of Sketch Engine itself: Manatee is a database manager system,

used especially for indexing of large corpora; Bonito, instead, is a web interface designed for Manatee that allows users to access the corpus search. Manatee is written in C++ although an API for other programming languages, e.g., Python, Java, Pearl, and Ruby is offered. Bonito is written exclusively in Python (Rychlý, 2007). Currently, the manager supports corpora in more than 90 languages (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004).

Sketch Engine is refreshing for a wide variety of reasons, among which there is the feature used in this thesis to retrieve the adjectives and modifiers of a certain word. Particularly, the feature before mentioned called 'Word Sketch', bases the results on collocations and word combinations. This feature, that Sketch Engine makes available, is rather vital. There is most importantly the possibility to download a series of adjectives from the internet and carry out the exact same experiment of this thesis without taking collocations and combinations into consideration. However, it would not have any sense: the metaphors in the previously described dataset are highly conventional, some more than others; in any case, they appear with a given frequency in humans' everyday language. This has consequences on the type of adjectives that could be used to describe the metaphor. Hence, it is important that the list of adjectives is first of all related to a certain word, and that the list is based on the collocations and combinations. Let us say that we may want to analyse the combinatorial behaviour of the word 'team' (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004): on Sketch Engine, we would select the corpus of interest, go on the page of Word Sketch and write the word 'team' on the search bar; Sketch Engine will take care of the rest and simply return the results we may be looking for. The results are arranged in categories, and it is possible to visualise only certain categories, for instance 'modifiers' and 'adjectives'.

Figure 2. Results of the search for team in Word Sketch, taken from Sketch Engine. Source: www.sketchengine.eu

As can be seen from Figure 2, the columns represent the categories under which all words related to 'team' can fall. It is possible to visualise only two or three categories, those in which we are particularly interested by clicking on the X on the top-right corner of each column.

In addition, the creation of Word Sketch files is based on a given corpus. Since metaphors are heavily pervasive and frequently used by individuals in texts or written formats from which corpora are built, it is important that the list of adjectives taken into consideration holds an account for word distribution in corpora.

A great possibility that Sketch Engine offer is that of downloading the search's results in different formats, according to the researcher's needs. In particular, for this thesis, the results of Word Sketch were downloaded as a Comma-separated Values (CSV) file, which was the easiest file to read and treat afterwards among all the other formats. One note should however be made: once the file is downloaded, even though only certain categories were selected on the website, Sketch Engine will insert in the downloadable file all related lexemes. Therefore, it may be necessary to modify the file, either manually, or with an appropriate function on Python, as it was done for this project.

2.2 WordNet

2.2.1 A Database of sense relations

When machines have to approach natural languages as speakers do every day, they need to possess information about the words used and their meanings, because "meaningful sentences are composed of meaningful words" (Miller G. A., 1995, p. 39). Consequently, it is indeed necessary that machines require some database they can use in order to have an 'understanding' of what meaning means to humans.

One of the most widespread and biggest sources used to consult lexical relations is WordNet (Fellbaum, 1998). The idea behind WordNet was that of understanding the learning process in children, and simulate it (Miller & Fellbaum, 2007). This attempt was inconclusive, since children are very efficient learners; however, it led to the discovery of interesting relations between and among words. Nowadays, WordNet can be seen as a large lexical database of English, where the four main content-word-related parts of speech (PoS) – namely, verbs, adjectives, adverbs and nouns – are grouped into cognitive synonyms, called 'synsets' (Miller, Beckwith, & Fellbaum, 1990). The synsets are connected with one another, with each synset expressing a different concept.

The WordNet database can be mistaken by a particular type of thesaurus, since it essentially groups the words based on their meanings. The similarity, however, stops here. WordNet adds some other features, like, for instance, the fact that not only the words are linked together, but also some specifc word senses. "As a result, words that are found in close proximity to one another in the network are semantically disambiguated" (Miller, Beckwith, & Fellbaum, 1990). The second great difference between WordNet and a thesaurus is that a thesaurus groups words without following a specific pattern other than similarity in meaning, while WordNet also labels the semantic relations among words.

WordNet uses synonymy as main relation among words. Therefore, words like 'shut' and 'close', or 'car' and 'automobile' (Miller, Beckwith, & Fellbaum, 1990) will be semantically related. So, technically it is fair to say that the synsets are groups of synonyms. WordNet at this time contains 117 000 synsets, and as a part of the relations with other synsets, each of them contains a "gloss" (Miller, Beckwith, & Fellbaum, 1990), which can be defined as a short definition of that synset.

In this database, the most frequent relations among synsets are the super-ordinate relation, called hyperonymy, and the subordinate relation called, hyponymy. For instance, the synset –

or better, hypernym - {bed} has many hyponyms, e.g., {bunk bed}, {couch}, {hammock}, etc. Quite obviously, {bed} will be an hyponym itself, in this case of {furniture} or {piece of furniture}. This reasoning can be carried out for all synsets in WordNet. The interesting side of these relations is that ultimately there will remain only one synset, which includes all the others, namely {entity}.

There is, however, another relation which is instantiated among synsets, that of meronymy, which is the relation between the part and the whole. In this case, a synset like {chair} will be related through meronymy to {back}, {seat} and {leg} (Miller, Beckwith, & Fellbaum, 1990). Interesting is that meronymy can be inherited by hyponyms. Consequently, since {armchair} is an hyponym of {chair}, it is going to be related through meronymy to {back}, {seat} and {leg} as well.

Just like nouns, verbs are grouped into hierarchies as well, called trees or troponyms. The lower the position in the tree, the more characterising will be the verb. As to say, if at the top of the tree there is the verb {communicate}, at the bottom of the tree there is {whisper} (Miller, Beckwith, & Fellbaum, 1990). Comparing the two verbs, {communicate} certainly expresses some features of a certain event or behaviour; however, {whisper} gives away more details about the way the speaker is behaving.

Finally, as far as the last PoS represented is concerned, namely adjectives, they are related through antonymy. A pair of antonymic adjectives would be good-bad, tall-small and so on. Each of these polar adjectives is then linked to other adjectives, which are semantically similar (Miller, Beckwith, & Fellbaum, 1990). Therefore, if the adjective 'good' is taken in consideration, some other linked adjectives may be: great, keen, neat, satisfactory, solid, and many others. WordNet distinguishes between two types of adjectives: ascriptive, i.e., which convey attributes and are therefore organised in terms of antonymy or synonymy; and nonascriptive, which can be seen as variants of nouns modifying other PoS (these will redirect to the corresponding noun synset file),

Two decades after the release of this database, WordNet established itself as a reliable tool for various NLP tasks, among which there are information retrieval and machine translation, since they both deal with Word Sense Disambiguation (WSD) (Miller & Fellbaum, 2007). However, its effectiveness in WSD is rather limited because its arcs are sparse. Fellbaum and Miller (2003) tried to address this issue by including "morphosemantic links" (Miller & Fellbaum, 2007, p. 211). If more connections are added, as a consequence more information of a certain meaning is given, which may result particularly useful for users.

3.2.3 Using WordNet

After retrieving the adjectives, the main issue was how to store them and use them. What became obvious by looking at the lists was that the adjectives needed a form of filtering. As a matter of fact, from the CSV file downloaded from Sketch Engine, it is possible to find not only adjectives, but also nouns that relate in one way or another to the words it is looked for. To name a few of the possible categories, and always taking the word 'team' from previous examples, it is possible to search for: nouns modified by 'team', modifiers of 'team', verbs with 'team' as subject, verbs with 'team' as object and so on (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004) (cf. Figure 2)
Although this may have some practicality to other researchers or to certain types of research, for this study the vast majority of the options were redundant. Hence, it was necessary to find a way to make the function consider only descriptive adjectives or modifiers. This was reached through a form of filtering, where the related words to the source passed through a set which discerned between descriptive and non-descriptive adjectives.

The second issue regarding this topic involved the method to use in order to accomplish this adjective filtering. After careful considerations, the safest choice included WordNet (Miller, Beckwith, & Fellbaum, 1990), since the database offers the possibility to distinguish between two different classes of adjective (cf. 2.2.1 A Database of sense relations) and therefore through NLTK (Loper & Bird, 2002), it would have been possible to filter the adjectives retrieved from Word Sketch, by selecting one of those classes. Secondly, WordNet is robustly reliable, in terms of relations between synsets; hence, if we want to be sure whether a certain adjective is relational or descriptive, the database can give a trustworthy answer. The last reason for using WordNet while filtering the adjectives involves the fact that WordNet was constructed manually. It is, hence, very unlikely that some mistakes regarding classification or misrepresentation of synsets are to be found in the database, errors that could be present if the database were collected automatically by a machine.

The first step was that of eliminating from the list of related-lexemes words that were not adjectives or modifiers; the second step was that of filtering all those adjectives that were not descriptive through a set written through Python. Let us make an example: in the noun *whale shark*, whale is technically a modifier, or better a noun that changed class to modify another noun. However, it is not a properly said adjective. The main reason for this being the case is that the word *whale* in the example is not nearly as descriptive as a real adjective may be, for

instance in comparison with *an enormous shark*. In addition, the theory at the base of the experiments does not require non-descriptive adjectives. Since the goal is to find features of the source domains (represented by adjectives) that are mapped on to the target domain, it would be redundant to consider non-descriptive adjectives as well, given their reduced descriptive nature. Therefore, the set created through NLTK needed to eliminate those adjectives that were not considered descriptive.

The set was created with few lines of code and three for loops. The idea bases on the fact that WordNet allows a differentiation between different adjectives based on their construction and not only can we access said adjectives, but also satellite adjectives. In WordNet (Miller, Beckwith, & Fellbaum, 1990) adjectives form clusters, which can have one or more head synsets, with these representing the antonymous relationship previously explained. An adjective satellite is a synset contained in each head synset (Miller, Beckwith, & Fellbaum, 1990).

The first step was that of creating an empty set at the beginning of the code, so that all the adjectives with said properties on WordNet could be stored in one place. The second step consisted in establishing an iteration over adjectives ('a') and satellites ('s'), so that all redundant pars of speech, e.g., nouns, verbs, and so on, would be left out from the set. The third passage was to furtherly iterate over all synsets in WordNet that were classified with those PoS, namely 'a' and 's', since a further filtering was needed to avoid non-descriptive adjectives to be inserted. Through this second iteration, only the lemmas were selected from the said synsets, to make two if statements possible. Finally, the set was led behaving in the following way: if the adjective received was relational – namely, if it derived from a noun – it was bypassed by the code, otherwise it was inserted in the set previously defined. To make an example of the differences between the two lists – namely, filtered and unfiltered – Table 1 shows the first twenty adjectives of each list.

Filtered adjectives	Non-filtered adjectives
Yellow	Fritillary
Such	Monarch
Purple	Hairstreak
Scarce	monarch
Endemic	100m
Likely	200m
Good	Swallowtail
Free-flying	peacock
White	Peacock
Mimetic	50m
Marbled	Bird
Colourful	Fritillary
Endangered	Moth
Migrant	Brimstone
Blue	Marsh
Tropical	Bee
Beautiful	Admiral
Rare	Tortoiseshell
Brown	Skipper
Orange	Marbled

Table 1. Comparison between the first 20 adjectives of the filtered and unfiltered lists

In addition to retrieving the adjectives associated to the source domain of each source-target pairs, it was also necessary to store the filtered adjectives in a repository, so that the function could easily select only the source in question and calculate the similarity. In order to do this, the Python's dictionary possibility was the most obvious choice.

Thus, for each source domain representing the entry, all the descriptive adjectives related to that source were grouped in one place. It is mainly appropriate because it makes the retrieval of the adjectives easier and smoother. This is particularly true in the case that the function is

inserted in a bigger model or in a pipeline: as a consequence, the timing executing the function will be shorter since every passage becomes automated.

WordNet (Miller, Beckwith, & Fellbaum, 1990) was not only used for the NLP side of the project, but also for the CV part, which may appear weird. However, ImageNet, the database from which images representing targets and sources were downloaded to train the NN, bases its structure on WordNet. With NLTK (Loper & Bird, 2002) comes a useful function that allows to take advantage of this relation: indeed, once the name of a given archive on ImageNet is known, it is possible to retrieve the corresponding synset on WordNet (if the archive exists). The function on NLTK is wn.synsets_from_pos_and_offset, where it is necessary to specify what PoS we are looking for as first argument, and the number present in the archive name as a second argument. The function will retrieve the corresponding synset and print it on screen (cf. Figure 3).

```
>>> wn.synset_from_pos_and_offset('n', 4543158)
Synset('wagon.n.01')
```

Figure 3. Example of usage of function 'synset_from_pos_and_offset'. Source: www.nltk.org

Thus, once it was checked which of the sources and targets were represented with an archive on ImageNet, a loop exploiting the NLTK function was executed, in order to download only the archives needed.

2.3 FastText

2.3.1 The Model

It has been indirectly established so far that metaphors deal with meaning: without a cognitive representation (and perception) of time and money separately, it would be difficult to interpret a conceptual metaphor like TIME IS MONEY.

Computationally speaking, however, representing meaning as humans supposedly do in their brains is quite difficult. Nowadays, the best way to carry out this task is that of using Distributional Semantics (DS). DS can be seen as a "usage-based model of meaning" (Lenci, 2018), where the distribution of linguistic units withing a given sentence or text is supposed to play a very relevant role. DS is compelling and widely used, and the reason is twofold: on one hand this type of model allows to represent meaning in a space – i.e., a vector space, this is the reason why DS is also sometimes known as vector space semantics; on the other hand, it makes use of computational methods to learn patterns and behaviours from said representations of linguistic data. It is also important to mention that DS is highly dependent on corpora, given the fact that in order to build these semantic vector-based models and then use them, it is necessary to obtain large and extensive linguistic dataset. Indeed, the position of the lexemes (which are vectors) in the vector space depends on the lexeme's co-occurrences in its context. Hence, it is through corpora that the co-occurrences are made available.

DS bases its representations on a theoretical foundation known as the Distributional Hypothesis (DH). The hypothesis is extremely simple, yet effective: "lexemes with similar linguistic contexts have similar meanings" (Lenci, 2018) and derives from previous work done by Harris (Harris, 1954) DS essentially allows to operationalise the DH, encoding the lexemes' properties of distribution into vectors. The choice for vectors fell because of they have geometrical interpretations. For instance, vectors have *n* components, which define a series of points in a *n*-dimensional space; thus, the distributional semantic representations are simply geometrical representations of the said lexemes inserted in a distributional (vector) space (Lenci, 2018).

The aim of DS is that of creating Distributional Semantic Models (DSMs), which consist in a configuration of the features required to build the distributional representations: lexemes, type of context, weights, dimensionality reduction, and similarity metric.

There are many different ways DSMs can be approached, and as a result various types of DSMs have been created (note to see extensive explanation). The choice of which type of DSM to use

in a given task, heavily depends on the desired outcome. For this particular case, it was necessary to have a pre-trained model, which represented tokens in a distributional space, was reliable, and computationally cheap in the first place. The model that possessed all these features was fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016).

Indeed, fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016) implements the efficient method at the base of Bag of Words (Bojanowski, Grave, Joulin, & Mikolov, 2016), which has, however, some backwards: if the number of classes is considerable, the computation of the linear classifier becomes rather expensive, computationally speaking. Therefore, to decrease the running time, the researchers used "a hierarchical softmax" (Joulin, Grave, Bojanowski, & Mikolov, 2017, p. 428). Softmax alone is a generalisation of the logistic function; however, its training times are rather long, and this makes the implementation of the softmax function quite difficult (Goodman, 2001). Hence, Goodman (2001) suggested a hierarchical softmax, where "each node is associated with a probability, that is the probability of the path from the root to that node" (Joulin, Grave, Bojanowski, & Mikolov, 2017, p. 428). In less complex words, this implies that the probability of each node is always lower compared to the probability of their 'parent'. Because the probability will always be lower, this reduces the training times, and this has positive consequences when models like fastText are used withing bigger models or more in general tasks.

As far as its functioning is concerned, fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016) represents a document with an average of word embeddings and then feeds this document into a linear classifier (Xu & Du, 2019).

Going a little bit more mathematical about this model, its architecture could be represented as follows: with a corpus $D = \{(d_i, y_i)\}$ for text classification, a document $d \in D$ is expressed with $d = (x_1, x_2, x_3, \dots, x_N)$ – x_i is an n-gram feature which can be found in the document (Xu & Du, 2019).

The n-gram features will subsequently be embedded in a "H-dimensional distributed vector representation" (Xu & Du, 2019, p. 1715). In order to make the representation of fastText easier, and have visual feedback, an image about the architecture of the model will be inserted below (taken from Xu and Du (2019, p. 1715)).

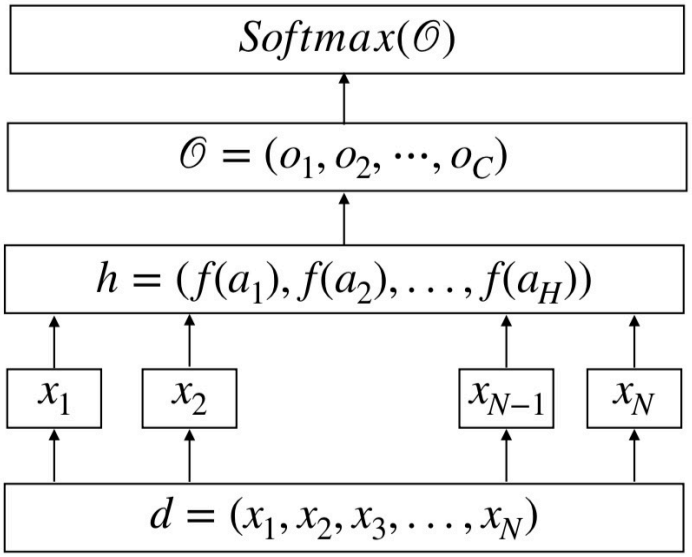

Figure 4. The architecture of fastText: f(a_1) is the average operation over the jth dimension in word embeddings

There is a small issue in fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016), namely the fact that the implementation of this model does not consider word order information, it actually discards it. The model sums the scores of each word, and consequently the word or more than one word which have a high absolute value will have a say in the final decision. It is possible however to insert the use of a bag of n-grams, instead of a normal bag of words. An n-gram is one of the simplest language models available, and it calculates the probability of a word based on the previous *n* words. Other very successful models which do consider, the sequence information, e.g., BERT (Delvin, Chang, Lee, & Toutanova, 2019), have demonstrated how valuable sequence information really is. The theory that word-order information is precious is confirmed by the fact that if fastText is implemented with bigrams, its performance increases slightly (Xu & Du, 2019).

The fastText model (Bojanowski, Grave, Joulin, & Mikolov, 2016) does not consist of one single word vector for English, for instance, but on its website, it is possible to find all word vectors in different languages and of different sizes. To make it practical, under Resources there is a section called 'Word vectors for 157 languages', which is quite self-explanatory of what can be found in there. These word vectors were trained on Common Crawl – an organization that crawls the web and publicise the resulting data (Grave, Bojanowski, Gupta,

Joulin, & Mikolov, 2018) – and Wikipedia using fastText (Word vectors for 157 languages, 2016). All the models can be downloaded either from a terminal or through a python script.
It is also possible to adapt the dimension: for instance, for the word vector of English, the pre-trained word vectors have dimension 300; however, if less dimensions are necessary, it is possible to just reshape it.

2.3.2. The role of context, distributional spaces, and fastText

One of the biggest questions for this thesis, which required a well thought-through answer, regarded how to represent the metaphors in a computational way.
As it was previously depicted, there have been an adequate number of studies regarding metaphors from different points of view: from psycholinguistics to cognitive science, to theoretical linguistics, and also NLP.
As far as semantic representation is concerned, when it comes to metaphors the computational methods typically seek to create some mechanisms, which can identify in one way or another the transference of properties, from one source domain to its target domain (Shutova, 2015).
There have been several different approaches to represent metaphors, going from logical representations to data-driven approaches; however, the technique that has had the most significant development in the recent years involves distributional semantics techniques (McGregor, Agres, Rataj, Purver, & Wiggins, 2019).
The distributional semantics' approaches derive from the theoretical work of Harris (1957), who advanced the observation that words which occur together in a certain context, are likely to be related in meaning. Therefore, computational models are nowadays trying to capture this feature by transforming words into vectors, and by placing them into high-dimensional spaces, called *vector spaces*.
When the question of how to represent metaphors arose, the most straightforward and reliable option was the usage of vectors. At that point the only issue was understanding how they would have been used, yet this will be discussed more into details with respect to the two experiments.

Once the choice the theory's approach of metaphor's treatment fell on vectors, a long discussion on which vectorisation model to use for the representation of metaphorical domains was carried out. The first choice had to be made between static and contextualised models. A

static model (e.g., fastText) represents a given word always with the same vector, independently from the context. Compare for instance a) and b):

a) I can't believe my parents just sold their **house** of 30 years.
b) This **house** was built in 1934.

The difference between a static and a contextualised model is simply that 'house' would be represented with the same vector in both sentences if we have a model like fastText, or with two different vectors – because they are indeed two different occurrences of the word – in a contextualised model, for instance BERT (Delvin, Chang, Lee, & Toutanova, 2019).
The problem was trying to find a model that was robust, reliable but at the same time fast enough. Because the function may be implemented in a bigger model or in a pipeline and needs to return a result rather fast, the speed of the vectorisation model was fundamental. After carefully pondering the possibilities, the choice fell on fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016) because of its reliable and relatively fast activation, but also for the purposes of the project. Since the work involves creating a pipeline that could potentially be used to interpret visual metaphors, using contextualised models like BERT (Delvin, Chang, Lee, & Toutanova, 2019) would have been difficult: if a machine is fed with a given visual metaphor, there is no context, the metaphor is exclusively represented in the combination of two images representing the source and target. The fact that in order for humans to process and understand a visual metaphor it is necessary to go through language and solve the conflict has no connection to the way BERT (Delvin, Chang, Lee, & Toutanova, 2019) works. Hence, a static model like fastText perfectly fitted the purpose.
The only way fastText was used was to retrieve the vectors for the adjectives and the target. The NLP side of the project involved two experiments, which involved two completely different approaches, as will be pointed out in the next sections; however, the usage of this model remained the same in both cases.
From fastText website (Bojanowski, Grave, Joulin, & Mikolov, 2016) the model chosen was the English version of the "Word vectors for 157 languages" (Bojanowski, Grave, Joulin, & Mikolov, 2016).
The model – like all the other 156 for additional languages – was trained on Common Crawl and Wikipedia (Bojanowski, Grave, Joulin, & Mikolov, 2016). The character n-gram length equalled 5, plus a window of size 5 and 10 negatives were added. Finally, the model was trained using CBOW with position-weights. CBOW is a model designed by Mikolov et al. (2013),

which “learns word representations by predicting a word according to its context” (Mikolov, Grave, Bojanowski, & Puhrsch, 2018, p. Model Description section).

It was downloaded through Python code, as indicated on the website. While it is possible to modify the model, for instance by adjusting the number of dimensions – which standardly is set at 300 – the dimension value remained unmodified. Since this project and the subsequent survey are purely explorative at this point in time, it would have been non-sensical to modify the hyperparameters and analyse further results; thus, no change was brought to the said model.

2.4 ImageNet

2.4.1 The largest database of images

With the advancements in technology and personal devices, images seem to have gathered attention and fame among both researchers and Internet users. In the most developed countries, every person has a smartphone with high-performance cameras – sometimes more than three cameras on one smartphone – to grasp every angle of the scenario in front of our eyes. Users post their photos online, given the fact that social media have been increasingly popularised. Therefore, the Internet is saturated with pictures from all around the globe. For instance, Google's Image Search (now Google Images) was launched in 2001 with a collection of 250 million images. In 2005 it reached 1 billion images, while by 2010 the amount rose to 10 billion, and it is still counting (Siegler, 2010).

At the same time, ML started developing together with the need for machines to recognise, classify and detect elements in images, in order to 'understand' them. The next quite natural step was that of creating algorithms that could exploit these large databases of images. There is, however, a problem with datasets like Google Images: it regards how this data is going to be used and organised (Deng, et al., 2009). Here is where ImageNet made its entrance.

ImageNet can be described as a "large-scale ontology of images" (Deng, et al., 2009, p. 248), on which researchers can rely to train and test algorithms for CV. The interesting feature of ImageNet is the fact that its hierachical structure is based on top of WordNet (see 2.2.1 A Database of sense relations). Therefore, ImageNet aims to provide averagely 500-1000 images to represent the synsets (Deng, et al., 2009). The images are organised in different classes contained in a semantic hierarchy, and the semantic structure is taken from WordNet. The synsets of images in this database are linked to each other by different types of relations, the most useful one being "IS-A". A great challenge when classifying the images while building ImageNet was the position on hierarchy: as the creators of ImageNet underline, the lower the position in the hierarchy, the more complex the classification will be; if there are two images, one of a Siamese cat (cf. Figure 5) and the other of a Burmese cat (cf. Figure 6), the differences between the two are not many, aside maybe the fur's colour, which makes the distiction hard (Deng, et al., 2009).

Figure 5. Picture of a Siamese cat. Source: en.wikipedia.org

Figure 6. Picture of a Burmese cat. Source: en.wikipedia.org

The construction of ImageNet followed a strict schedule, since the project was rather ambitious. The first step was that of collecting the images for each synset in WordNet. Once the dataset of images was collected from the Internet, the creators had to clean the candidate images, for which they relied on humans. Individuals were retrived through Amazon Mechanical Turk (AMT), a platfrom appropriate for labelling tasks on a large scale. The Turkers were presented a set of images for a synset, plus a definition and a link to Wikipedia: their task was that of verifying that every image contained the object mentioned in the synset. Because Turkers are anyway humans, and because humans make mistakes and do not follow instructions properly at times, ImageNet's authors considered only those images which

obtained the majority of positive votes (Deng, et al., 2009). The last step, was that of inserting the labelled images in a dataset, which is now available online (ImageNet, 2021). The small subset which can be used for CV tasks, e.g., object recognition, can be downloaded without any particular request, and it contains 1000 classes. Upon permission granted, it is possible to download the database.

2.4.2 The challenges launched by ImageNet

Once Fei-Fei Li and other collaborators (Deng, et al., 2009) created the dataset of ImageNet, every year they proposed a challenge called ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Brownlee, 2019). The challenge is well known in the field of CV and has become a "benchmark for large-scale object recognition" (Russakovsky, et al., 2015, p. 211). The idea behind this challenge comes from another challenge, namely the PASCAL VOC (Everingham, Gool, Williams, Winn, & Zisserman, 2012), which started in 2005. Both competitions consists in two segments. The first component is represented by a dataset that is publicly available, while the second one is an annual challenge, followed by a workshop.
The annotations in ILSVRC can be assigned to two categories: image-level annotation and object-level annotation. The task, most simply, involves predicting the content of pictures in order to annotate them automatically. Over the years, however, the tasks become more specific, to adapt to certain needs. Russakovsky, et al. (Russakovsky, et al., 2015, p. 213)suggest an overview as follows:

> " (1) *Image classification* (2010 – 2014)
> (2) *Single object localisation* (2011 – 2014)
> (3) *Object detection* (2013 – 2014)"

Image classification simply predicts the classes for the objects which can be found in the picture. Single object localisation involves image classification and the task of drawing a rectangle around only one example of the objects present. Finally, object detection deals with, again, image classification and the task of drawing a rectagle around each object in the picture.

This division was up-to-date until 2015; yet, with the great results obtained in the first 5 years, the tasks became recently more complicated, e.g. video labelling – i.e., not simply pictures.
Every year, a training dataset was released together with a test dataset which was instead not annotated. The duty was that of making annotations for the test dataset and submit them to an evaluation. As far as the sizes are concerned, usually the training set contained ca. 1 million images. Instead, the validation and test sets were comprised of ca. 50,000 pictures and ca. 150,000 pictures respectively (Brownlee, 2019).
The ILSVRC has become a milestone in CV field, and more in general in artificial intelligence. Since the first publication in 2010, the improvement's rate has been striking, with more and more researchers joining the challenge as years went by. In 2010, ILSVRC counted 35 entries, while 6 years later the entries were 172. In 2017, the last time ImageNet organised the challenge, the number of entries diminished to 115, which is anyway higher than the early entries. As it can be imagined, with the enhancement of diverse neural networks, the classification error rate underwent some changes as well. In particular, in 2010 the error rate was at 0.28, it significantly decreased in 2014 to 0.07, to reach the 0.023 in 2017. The localisation error followed the same path: in 2010, the rate was 0.43; in 2014, it was equal to 0.25; and in 2017 it decreased to 0.062 (Russakovsky, et al., 2015).

The success gained over the years, has been mainly achieved through deep neural models, namely the convolutional neural networks (CNNs).
The first neural network that managed to achieve highly successful results was AlexNet (Krizhevsky, Sutskever, & Hinton, 2012). The researchers from Toronto University developed a CNN which achieved the best results both in the ILSVRC-2010 and ILSVRC-2012. The architecture of Alexnet containes eight layers: five of these are convolutional layers, while the remaining three are fully-learned layers. The unusual feature of this neural network is the usage of the Rectified Linear Units (ReLUs). Standardly, neuron's output *f* is modelled either through a hyperbolic tangent $f(x) = tanh(x)$ or a sigmoid function $f(x) = (1 + e^{+x})^{-1}$. Nair and Hinton (2010) suggested, however, another function, namely the ReLU, that could address the slowness of the other functions to get an acceptable error rate. Indeed, a CNN with four layers using a ReLU on CIFAR-10 managed to reach a 25% or error rate in training six times faster compared to the same CNN using the hyperbolic tangent (Krizhevsky, Sutskever, & Hinton, 2012). Furthermore, the kernels in the second, fourth and fifth convolutional layers were connected exclusively to the kernels in the previous.

AlexNet's researchers had to cope with some issues regarding overfitting, i.e., a situation where the model fits perfectly the training data, and is not able to generalise (and therefore perform well) against unseen data. The neural network had 60 million parameters; however, even though the ILSVRC datasets contains generally 1000 classes and obliges to some constraints while mapping the labels, this was not enough to learn 60 million parameters without going overfitting. Therefore, the computer scientists adopted two measures: firstly, they enlarged the datest using "label-preserving transformations (e.g., [25, 4, 5])" (Krizhevsky, Sutskever, & Hinton, 2012, p. 5); secondly, they used the "dropout", which consists in assigning the value 0 to the output of hidden neurons, when its probability is 0.5. This method seems particularly efficient because it combines different models' predictions without weighting too much on the training time, and at the same time it reduces the error rate in the testing phase.

As far as the results of AlexNet are concerned, for the ILSVRC-2010 they reported to have achieved "top-1 and top-5 test set error rates of 37.5% and 17.0%" (Krizhevsky, Sutskever, & Hinton, 2012, p. 7). The best results by other neural networks during the challenge were 47.1% and 45.7% for top-1, while 28.2% and 25.7% for top-5. In the ILSVCR-2012, the comparison are difficult to make, since the test set labels were not available. What is known is that the CNN described in Krizhevsky, Sutskever & Hinton (2012) achieved a top-5 error rate of 18.2%.

The second big leap in the state of the art was made in 2014 by GoogLeNet (Szegedy, et al., 2015) during the ILSVRC-2014. GoogLeNet is a neural network based on the Inception architecture and models, where all convolutional layers are paired with the ReLU non-linear activation function. The interesting feature about this neural network is its practicality and efficiency, since it was built so that the inferences could be run on individual devices (Szegedy, et al., 2015).

During the challenge, the researchers trained seven versions of the same GoogLeNet network independently: the only features that differed among the seven were the sampling methods and the input image order, which was obviously randomised. The final submission for the classification challenge registered a top-5 error rate of 6.67%, resulting to be the first network for ILSVRC-2014. GoogLeNet achieved the first place in the detection challenge as well, with a object detection mAP (%) of 43.9 – in this section, the second place was assigned to CUHK DeepID-Net and the third was assigned to Deep Insight, with a mAP (%) of 40.7 and 40.5 respectively.

The last neural network it is worth mentioning – also because it will be used in the CV side of the project – is ResNet (He, Zhang, Ren, & Sun, 2015). The state of the art had been well established by previous deep convolutional neural networks, which made huge advancements

in image classification possible. The idea behind the enhancements in this field can be summed up by this question: “Is learning better networks as easy as stacking more layers?” (He, Zhang, Ren, & Sun, 2015, p. 770). Although appealing, this question leads to one of the most known problems in CV, namely the vanishing and/or exploding gradients (Hochreiter, 1991).

What has been noticed is that when deeper neural networks can converge, another issue appears, namely a “*degradation*” (He, Zhang, Ren, & Sun, 2015, p. 770). This rising issue is fundamental, because it shows that not all systems are easy to optimise. Therefore, the authors try to address the problem of degradation by creating the deep residual learning framework[3].

To develop a robust neural network that would use the deep residual learning, the authors created two types of networks: plain networks and a residual networks (which ultimately is the ResNet itself). The plain networks are based on the VGG nets (Simonyan & Zisserman, 2015). They created two versions of the plain networks, one which contained 18 weighted layers and another which contained 34 layers. In addition, the convolutional ones had 3×3 filters – fewer compared to the VGGs. The residual networks were based on the plain networks, but with the adjuction of shortcut connections. These connections turned the plain network into its counterpart, ultimately the residual version. The shortcuts can be used directly when the input and the output have the same dimension. If the dimensions increase, the network considers to cancel the projection step by performing an identity mapping.

In order to test the reliability of the two networks, and give an insight on their performance, He, Zhang, Ren & Sun (2015) tested both models on the ILSVRC-2012, which contained 1.28 million images for training, 50.000 images for validation, and 100.000 images for testing. For the 18-layer plain network, the top-1 error rate stopped at 28.54%., while for the 34-layer plain network, the value was at 27.94%. This counter-intuitive discrepancy led the authors to the analysis of the training phase, where they saw that the 34-layer network had higher training error, and this propagated to the testing phase too.

The ResNets did significantly better. What is interesting about a comparison between the two layers is that the situation of the plain networks is actually reversed: the 34-layer network performed better. In particular, the 18-layer network reported a top-1 error rate of 27.88%, while the 34-layer network attested a top-1 error rate of 25.03%.

Furthermore, to confirm the abilities of the ResNet, they tested the network on the PASCAL VOC 2007 and 2012 sets, where the network reported an object detection mAP (%) of 76.4

[3] The framework iteself will not be addressed in this thesis. For further information on the topic, please refer to the paper mentioned.

(VGG = 73.2) and 73.8 (VGG = 70.4) respectively (He, Zhang, Ren, & Sun, 2015, p. 777, Table 7). The same was done on the COCO validation set, and the network disclosed an object detection mAP (%) of 48.4 (VGG = 41.5) (He, Zhang, Ren, & Sun, 2015, p. 777, Table 8).

The most crucial milestones are the networks mentioned above. The ILSVRC posted two other challenges, respectively in 2016 and 2017; nevertheless, the three neural networks already presented supposedly decribe the advancements in the field.

3. THE PROJECT

3.1 The background

Machines seem to have a hard time processing anything related to human language, which is comprehensible given their construction. However, some aspects of natural language appear to be particularly difficult for machines, leading to terrible mistakes or results. One of these aspects is metaphors. As it has been previously well defined, metaphors are vividly pervasive in everyday language; therefore, it is important to find an implementable method that leads machine to a correct interpretation of the said elements.

It has been argued many times since the advancements in NLP that statistical methods and computational representation methods will never be able to sample a natural language. And the reason for that is the fact that natural languages are infinite (Chomsky, 1957). Therefore, it would be apparently impossible to tackle and make a 'finite' computer understand something infinite. This issue has been raised against NLP more generally, but also against distributional semantics and its models. Distributional semantics has been dragged into the feud because although it tries to build representations of words or sentences through vectors which respect their co-occurrences in corpora, they are not able to generalise their methods to unseen word combinations (Turney & Pantel, 2010).

Metaphors to a certain extent are unseen: conventional metaphors have now entered the vocabulary of everyday language; therefore, it is possible to study their occurrences, their frequency and structure. However, as it was previously delineated, not all metaphors are conventional, and sometimes individuals create new unseen metaphors. At that point the metaphor interpretation from a machine's point of view may be extremely difficult since there is no corpus or distributional space representing the relational mapping between the given source and target domains.

There have been sever attempts at solving the issue of unseen word combinations. One of these, which has eventually partially inspired the technique used in experiment 2 of this thesis, involves additive models and multiplicative models between two vectors (Mitchell & Lapata, 2008).

Given two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, Mitchell and Lapata (2008) identified two models among others:

- $\mathbf{p} = \mathbf{Au} + \mathbf{Bv}$ (Additive, where **A** and **B** are two matrices);

- $\mathbf{p} = \mathbf{Cuv}$ (Multiplicative, where $\mathbf{C}$ is a weight tensor) (Vecchi, Baroni, & Zamparelli, 2011).

The goal of Vecchi, Baroni, and Zamparelli (2011) does not involve metaphorical interpretation directly. In their paper, they described the attempt to "characterize the semantic deviance of composite expressions in distributional semantics" (Vecchi, Baroni, & Zamparelli, 2011, p. 1). While doing so, they also evaluate other compositionality models – among these the one by Mitchell and Lapata (2008).

The results of their project are rather preliminary; however, it is possible to notice some encouraging evidence. These suggest that "simple unsupervised cues can significantly tell unattested but acceptable [Adjective-Noun expressions] apart from impossible […] ones" (Vecchi, Baroni, & Zamparelli, 2011, p. 8).

The idea for the experiments – in particular experiment 2 – partially came from the review conducted by Vecchi and colleagues. The manipulation adopted did trigger to a certain extent the rationale behind the theory of the experiments: using a combination of adjectives and target to select the adjectives themselves. Nevertheless, this will be addressed later on.

The project exactly aims to create a simple, yet effective, tool that machines can use when it comes to understanding what a certain metaphor may represent. Although two paths were undertaken, as it will be later explained, the smoother one involved a function and vectors.

The foundational mechanism behind the project is that of mimicking in the plainest way the process of mapping metaphors, which currently happens only in the human brain. As it was clearly exposed in the first chapter, a metaphor simply consists in mapping some features of one concept into another, which normally does not possess. As far as this project is concerned, the scope is exactly that of applying the same mechanism to machine's metaphor interpretation. Thus, if the intent is that of reaching human comprehension of natural languages in machines, it is rather important that we try to copy what happens in a human brain during a certain task and implement it – with due modifications – to NLP models.

Consequently, the most suitable way to represent metaphors – and more generally meaning – to machines is by involving vectors. Vectors can have a high number of dimensions; they have different features that allow manipulations. Therefore, the theory bases exactly on this ability to play and combine vectors to represent a simple mechanism like the mapping involved in metaphor creation. In addition, vectors are represented by numbers, and computers work significantly better with numbers rather than words. Indeed, using vectors can be seen as a translation of meaning into machine's language, reason why this method is extensively used in distributional semantics.

The project underwent some modifications along the way, mainly due to different issues. The original idea was to examine a machine's interpretation of visual metaphors. Visual metaphors were interesting because not only do they possess features which make image classification quite difficult – i.e., the source and the target images are mixed, or sometimes one is absent – but they also deal with language, since humans interpret visual metaphors through the language they speak and because advertisers often insert text anchors. There was one major impediment with this idea. As a matter of fact, a well-made, equipped, and reliable dataset of visual metaphors does not exist; there was a project on creating a visual metaphor corpus called VisMet (Metaphor Lab, 2017) started by the Metaphor Lab in Amsterdam. The corpus is available on the website; however, it is far from complete, and some annotations are not clear, which makes the usage of the few visual metaphor present rather complex. Indeed, the total amount of visual metaphors collected is 353 images, and while this may be sufficient for an NLP metaphor interpretation, for an image classification task they are not enough. Machines require thousands of images on which they can train, and then return significantly good results. What is more, for some of the said 353 images, certain annotations are not inserted, e.g., the source of the image or the URL. This may not be a problem if the dataset were really extensive, since it would be possible to avoid those incomplete images; however, with such a small dataset, there is no space for deleting images.

For these reasons exclusively, the original idea was modified, and instead of considering visual metaphors as a whole, it only considered visual representations of the domains. In other words, if the source domain is MONEY and the target domain is TIME, the representations would be a clock (cf. Figure 7) or watch and a moneybag or paper money (cf. Figure 8) respectively.

Figure 7. Image of money, taken from ImageNet database (Deng, et al., 2009)

Figure 8. Image of time, taken from ImageNet database (Deng, et al., 2009)

This would technically be the second step in a pipeline for visual metaphor interpretation: once the machine is given the full image, it is supposed to recognise the image for the source and the image for the target (assuming both are present) and return the correct class. The first step is in itself rather complex to carry out. Many visual metaphors (cf. chapter 1.1.4) do not display both domains; one domain may be represented by the shape given to the first domain; for instance, in the above-mentioned Heineken's advertisement (cf. Figure 1), the CDs were placed one on top of the other to form the shape of a can of beer. In this scenario, it is peculiarly difficult for a NN to recognise that one domain is the can of beer – which is technically not present – and the other is the pile of CDs. Hence, since the step of feeding the machine with the original visual metaphor was not possible, the project started from the second step.

Once the project was reshaped, the first major problem encountered was how to represent the metaphors and their mapping systems to the machine. The choice fell on vectors, as it will be explained in the next section; nevertheless, using vectors limits the number of possible routes. Accordingly, the key to combine the current model on human metaphor processing and the use of vectors would be that of unifying them (with either multiplication or addition) as to modify the vector of a given target.

The second significant problem was collecting the images for the computer vision part. The choice fell on ImageNet (ImageNet, 2021), since the dataset is highly reliable, and the quantity of total images is beyond enough. However, because many sources and targets were abstract nouns, e.g., FRIENDSHIP or LOVE, the dataset of metaphorical pairs had to be reduced. Therefore, the compute vision side of the project is more of a proof of concept, which may be extended when more images are found.

This issue is not impeding the NLP side, nor it defines whether the theory works or not. Indeed, the first step is a mere image classification, and CV has made huge leaps in the task – as it was delineated in the previous chapter. Hence, whether the classification works or not is a CV problem, related to the choice of the NN or the dataset created.

This leads to a discrepancy between the dataset used for the NLP side and the one used in the CV task. The discrepancy addresses the number of couples, not the type of couples, which makes the pipeline possible anyway. As a matter of fact, the search for further images in order to fill the void would furtherly test the NN accuracy in correctly classifying the images; nevertheless, it would not have an impact on the machine's ability to interpret a given linguistic metaphor.

The NLP side of the project explored two possibilities. Experiment 1 based the search for adjectives on an intersection of lists; however, the results were totally inconclusive.

Experiment 2, instead, tested the optimality of creating a function to find the correct adjectives, which may interpret a given couple of source and target domains. The results of the experiment 2 were significantly better. Since the evaluation of the choices returned by the computer could not be subjective, a survey was conducted on Qualtrics. Since it was important to have a reliable and most importantly human judgement on how well the machine performed on a normally humans-related task, a survey was conducted on Qualtrics. The survey asked participants to choose at least one adjective, or if none were appropriate, they could insert one in a box.
According to individuals' opinions, the machine overall performed well since they always selected at least one of the given adjectives. Participants also inserted some other adjectives in the box, even though they selected some of the available choices. Nevertheless, out of the eighty-four questions, no participant inserted adjectives in the box exclusively. This leads to the possibility that for each metaphor presented to the participants, the machine always returned at least one – among the ten – appropriate adjective.

To sum up, the explanation of the project for this thesis will proceed as follows: in the first place, some more precise information about the dataset, the adjectives and the vector choice will be provided; next, experiment 1 and experiment 2 will be described more in details, giving an insight on the methods, materials, and experimental settings; then, the survey and the image classification task will be analysed in details; finally, the chapter will conclude by displaying the results and their analysis, and by furtherly discussing the bigger picture, that is the project.

3.2 The dataset, the adjectives, and the vectors

3.2.1 The metaphor dataset

The first big step was that of collecting the metaphors. After careful research, the final choice fell on a previous work by Alice Coli for her master's thesis project (Coli, 2016).

Coli's work is rather distant from this thesis' topic, treating a completely different field of linguistics. Her thesis involved two experiments with the intent on demonstrating the Metaphor Interference Effect (MIE). The MIE is a "response time phenomenon wherein judging whether metaphorical sentences are literally true or false takes significantly longer than judging control sentences" (Chouinard, Volden, Hollinger, & Cummine, 2019, p. 270). This effect was first studied by Glucksberg, Gildea and Bookin (1982). The authors wanted to investigate whether the written comprehension of metaphors in humans was due to a "serial or simultaneous processing" (Chouinard, Volden, Hollinger, & Cummine, 2019, p. 271). The experiment set-up to test the theory was quite simple; participants were presented four types of sentences: *literally true* (e.g., "Some insects are bees"), *metaphors* which were really metaphorical but false in meaning (e.g., "Some roads are ribbons" (Chouinard, Volden, Hollinger, & Cummine, 2019, p. 271)), *literally false* sentences (e.g., "Some trees are nurses" (Chouinard, Volden, Hollinger, & Cummine, 2019, p. 271)), and *scrambled metaphors* which were false and literal (e.g., "Some roads are princesses" (Chouinard, Volden, Hollinger, & Cummine, 2019, p. 271)). The participants' task was judging whether the sentence they were presented was really true or false.

According to the stages of written metaphors understanding, the integration of relevant information which is needed to generate literal or metaphorical meaning happens on stage 2 (Glucksberg, Gildea, & Bookin, 1982). Therefore, the idea is that if there were a simultaneous presence of both literal and non-literal meanings at stage 2, the simultaneous presence of judgement "true" and judgement "false" for metaphors would create the Metaphor Interference Effect. More specifically, it would be possible to observe longer reaction times for metaphorical sentences compared to literal sentences (Chouinard, Volden, Hollinger, & Cummine, 2019).

Coli (2016) and her supervisor Cristina Cacciari investigated the MIE by modifying the experimental paradigm used by Glucksberg, Gildea, and Bookin (1982), using two experiments. In experiment 2, instead of presenting participants with whole sentences, Coli (2016) presented couples of words. They used three types of couples: literal, scrambled literal,

and metaphorical. Participants needed to decide whether the items in a given couple belonged to the same literal category or not: the correct answer was "yes" for literal category, while it was "no" for the other two options – namely, scrambled literal and metaphorical.

Experiment 1 was instead an already carried out by Cacciari, Semeghini, and Leonardi (Coli, 2016). This first experiment is equivalent to experiment 2, apart from the usage of the Divided Visual Field (DVF), which was used in the second experiment but not in the first one. The DVF allows to present the first word in a given pair at the centre of the visual field, while the second word of the pair will be presented either on the right or the left. Because this technique was not used in experiment 1, all the items in every source-target pair appeared at the centre of the visual field, without lateralisation (Coli, 2016).

As far as the first experiment is concerned, results showed a significant activation of MIE, with the classification of metaphorical couples requiring longer reaction times compared to the class of scrambled literal pairs.

The metaphors used for this thesis were taken from experiment 2, for a total of seventy-four source-target pairs. Some of these were well-known and commonly used metaphors, e.g., TIME IS MONEY; while others were less common, but still quite easy to understand, e.g., AN ACROBAT IS A BUTTERFLY.

In order to counterbalance the stimuli and avoid interferences on the value of MIE, Coli (2016) also analysed the couples of words in terms of different variables. These were:

- *Frequency*: the frequency's value was based on how often the pairs would occur in everyday language. This was carried out through Google's search engine.
- *Association between the two words*: Coli (2016) asked 60 individuals to say the first word that came to their mind when they were told a specific one. This allowed to make sure the two words in the pairs were not semantically associated.
- *Familiarity*: this value is highly subjective since it depends on the person's exposition to a certain couple of words. Individuals could assign a value between 1 (not familiar) and 7 (highly familiar).
- *Goodness*: this value reflected how well the linking between the two words was expressed. Once again, participants could assign a value between 1 (badly expressed) and 7 (well expressed).
- *Innovativeness*: the 60 participants who analysed the materials had to assign, again, a value between 1 (not innovative) to 7 (very innovative) to the pairs.

- *Valence*: this value was estimated through a scale going from -3 (negative), passing to 0 (neutral), to +3 (positive).
- *Concreteness*: every word was assigned a value between 1 and 7. The pairs were considered concrete if they had a value lower than 3, while they were considered abstract if they had a value higher than 3.
- *Comprehensibility*: a survey was distributed to the 60 participants; the individuals had to estimate how comprehensible a given couple was, assigning a value from 1 (not comprehensible) to 7 (very comprehensible).

What can be observed on the final dataset of 74 metaphors, the pairs were all assigned a value between 3 and 7 as far as comprehensibility is concerned. However, only few pairs were assigned the highest value. It is fair to say that this behaviour is rather normal: with metaphors it is hardly ever possible to reach very high score in comprehensibility, since the mapping mechanism at the base of metaphors is not completely straightforward. The idea of metaphor has indeed the intention to disrupt to a certain point the concept of the target domain.

The second experiment in Coli (2016) returned that participants had longer reaction times when they had to classify a metaphorical stimulus compared to a literal stimulus. This is in line with the results of Glucksberg, Gildea, and Bookin (1982) and of experiment 1 (Coli, 2016). This thesis does not treat the MIE topic; however, describing the previous work by Coli (2016) was fundamental in order to understand the dataset used for this thesis. That being said, the description of Coli's work stops here.

Both the work and the dataset of 74 source – target pairs were in Italian, while this thesis was conducted in English. Therefore, a translation of the nouns constituting the sources and the targets was conducted. The meanings of the metaphors continued to remain constant from one language into another, hence no couple needed be eliminated from the list because the translation had made the metaphor non-sensical.

In addition to the Coli's dataset, further ten metaphors were inserted in the dataset, due to issues in finding images for the computer vision side of the project, as previously delineated. These ten metaphors were taken from the website leverageedu.com (Sidrah, 2021). The webpage contains the most common metaphors in English, ranked from easy to difficult. In particular, this website can be used in preparation for English Language certification, since the metaphors present in the page are often used in the Use of English section of the exam; however, it is useful more generally to students who want to improve and learn more English metaphors and proper expressions.

The additional ten metaphors chosen to increase the dataset were chosen from a standpoint of intuitiveness. Indeed, all of them were highly understandable, highly common, and most importantly highly comprehensible.

3.3 Experiment 1

3.3.1 Introduction

The first attempt at creating an approach for metaphor interpretation through vector was a small stretch. In order words, the simplistic features, and its linearity already foretold that there was a high probability that it would have failed. Nonetheless, because of its purity and simplicity, it was also worth making the attempt, because if it worked, it would have been a very easy method.

The first step in the description of this experiment needs to remind that the metaphors of the dataset retrieved for the thesis are highly conventional. This has a connection with the type of adjectives retrievable from the models (namely, fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016), and Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004)) used in this experiment.

Both fastText and Sketch Engine strongly base their word retrieval (or vector retrieval for the first model more specifically) on the co-occurrence of words. Therefore, the adjectives either retrieved from fastText's function nearest neighbours or through Word Sketch on Sketch Engine would always be influenced by the words they are connected to. In other words, when retrieving the adjectives from 'dog', it is very unlikely to find adjectives such as 'technological' or 'rotten' on the list. Hence, the choice of adjectives would not be random, or partially so, but it would have based on a certain given word, which was the source or the target.

This supposedly should have had an impact on the results for the first experiment, although the reasoning is somewhat faulty.

As it was previously asserted, since the adjectives are not random but linked to the source and target in a pair, the adjective that could possibly describe the given metaphor should be present in both lists – one for the source and one for the target – because, once again, the metaphors in the dataset are conventional.

Therefore, as it is possible to evince, the idea for the first experiment was based upon the fact that given the high usage of said metaphors the adjectives describing them would have appeared in the lists. As it will be furtherly explained in the results, the first experiment was highly inconclusive, since it was not possible to find a pattern in the subset of metaphors used to test the concept.

What is more, even if the results were not inconclusive and the idea actually worked, other issues would have manifested later on, when applying the approach to novel metaphors: since

they are novel, there is no vector space or Word Sketch that contains the occurrences of the word order appearing in the metaphor; it would also be impossible to choose a good adjective, because the metaphor has never been 'described'.

3.3.2 Methods

Since experiment 1 has been treated as a proof of concept, only a subset of the full dataset for the thesis has been considered, ready to be expanded in case the results were promising. The subset of the dataset consisted in six randomly chosen pairs (in the following list, the first noun is the target, while the second noun is the source, following the 'X is Y' format):

- Idea – prison (comprehensibility rate: 5.43)
- Word – razor (comprehensibility rate: 3.84)
- Party – hurricane (comprehensibility rate: 5.67)
- Alcohol – burden (comprehensibility rate: 5.67)
- Lawyer – shark (comprehensibility rate: 4.22)
- Concept – maze (comprehensibility rate: 5.43)

Although randomly chosen, all metaphors are highly comprehensible and often used. This characteristic should have, in theory, covered for the fact that no vector manipulation was applied, but simply the method relied on already made vector spaces.

As far as the methodology for achieving the results is concerned, it based itself on lists comparison and was rather simple, as previously mentioned.

The first step was that of downloading an adjective list for each member of the metaphorical pair: for instance, taking the first couple in the experiment, there was a list for IDEA and a list for PRISON. These lists contained all adjectives that frequently accompany the sources and the targets. The list retrieval was done for all six pairs, and what is more, was carried out both with Sketch Engine and fastText (through the 'nearest neighbor' function). This has, as stated, implications on the type of adjectives, since they would follow occurrences in corpora.

The second step was that of filtering the adjectives since it was necessary to eliminate all adjectives which could not be considered descriptive. For this step, the formerly Python set built with NLTK was used. The remaining filtered adjectives were inserted in lists, which were used in the following steps.
The third and final step was the crucial one since it would have tested the theory behind experiment 1. For each of the six couples randomly picked, the list of the source and the list of the target were intersected with each other, in order to return the adjectives that were common to both. Since the metaphors were conventional, at least one correct adjective was expected to be returned. This step was conducted both with the lists retrieved from Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004) and the lists retrieved from fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016).
The experiment, as foreseeable, was carried out for all six pairs.

3.3.3 Results

The results for experiment 1 did not undergo any particular analysis, as it happened for the second experiment. Indeed, the first experiment was resembled a proof of concept, and in addition, the return of adjectives could be interpreted without conducting further analysis: if the two lists both contained certain adjectives, these would be printed out on screen; otherwise, the list remained empty.
The experiment was to be considered inconclusive for different reasons. More specifically, the reason concerning the results is that since the metaphors were conventional, at least one correct adjective was expected to be present in both lists. However, this was not the case.
Out of the six pairs before mentioned, only the pair *concept – maze* contained an adjective that could describe the given metaphor and that was present in both lists – namely that of the source and that of the target. The adjective in question is ‘difficult’, and albeit simple, it does sum up the idea that the metaphor is trying to convey.
As far as the other couples are concerned, the adjectives in common, if there were any, were completely out of context and were not linked to the meaning of the metaphor. Here is a table with the six couples and the adjectives in common from lists retrieved from Sketch Engine.

METAPHORICAL PAIR	ADJECTIVES
Idea – prison	good, bad, different, more, such
Word – razor	straight
Party – hurricane	bad, strong, major, perfect, third, more
Alcohol – burden	due, excess, likely, excessive, high, such
Lawyer – shark	present
Concept - maze	easy, difficult, simple, mathematical, entire, complex, traditional, moral, legal modern.

Table 2. Adjectives in common between sources' lists and targets' lists retrieved from Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004).

Below, six Venn diagrams will be inserted for each pair with adjectives retrieved from Sketch Engine, to visually represent the data inserted in Table 2.

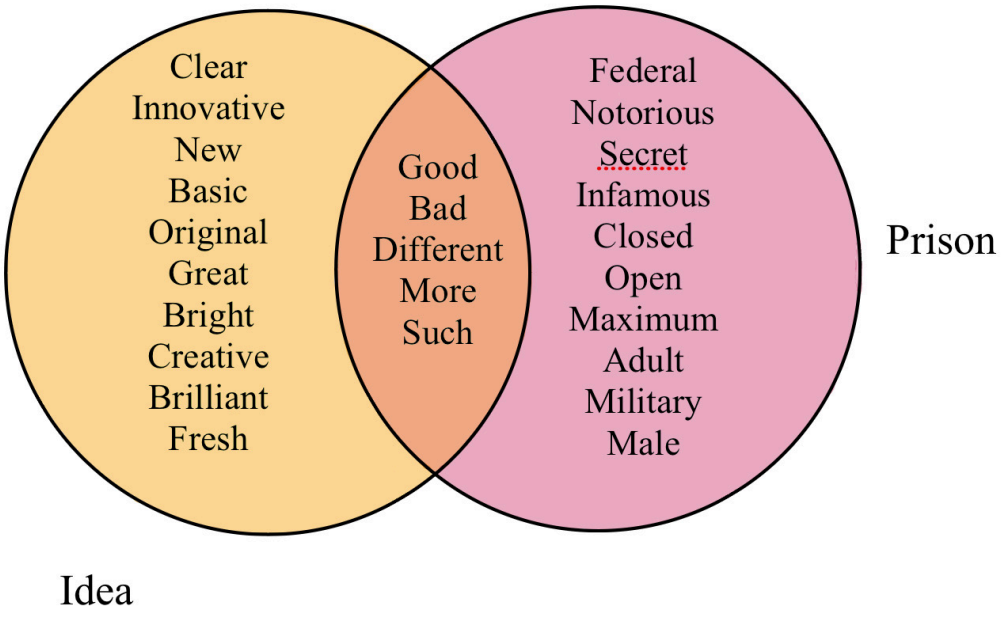

Figure 9. Venn diagram of adjectives retrieved from Sketch Engine (couple idea-prison)

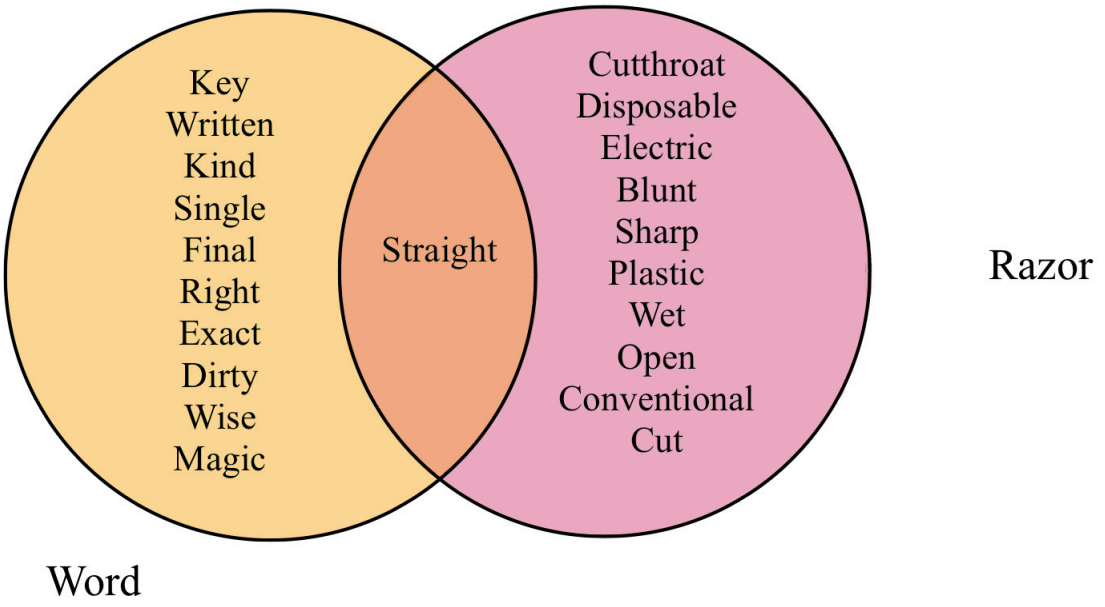

Figure 10. Venn diagram of adjectives retrieved from Sketch Engine (couple word-razor)

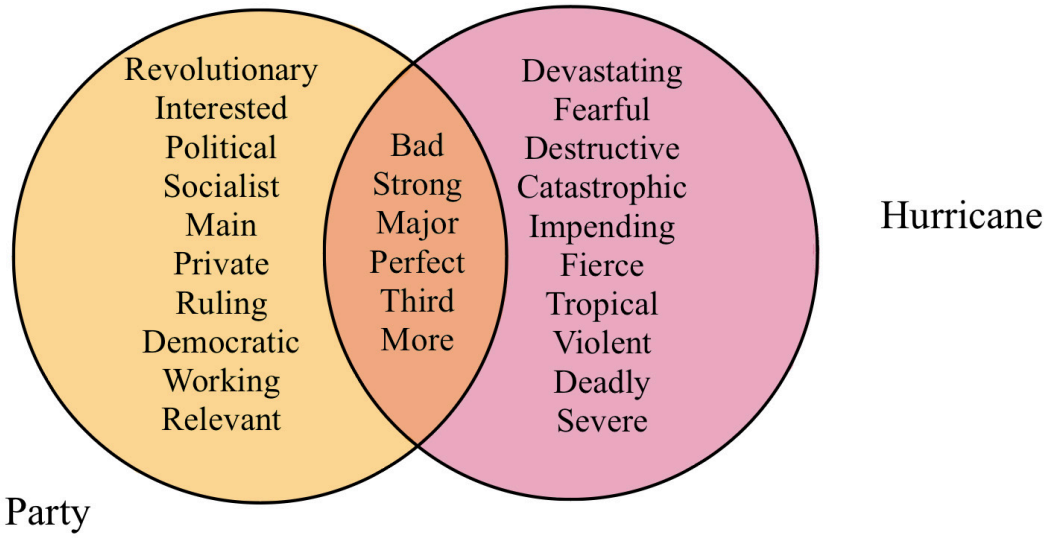

Figure 11. Venn diagram of adjectives retrieved from Sketch Engine (couple party-hurricane)

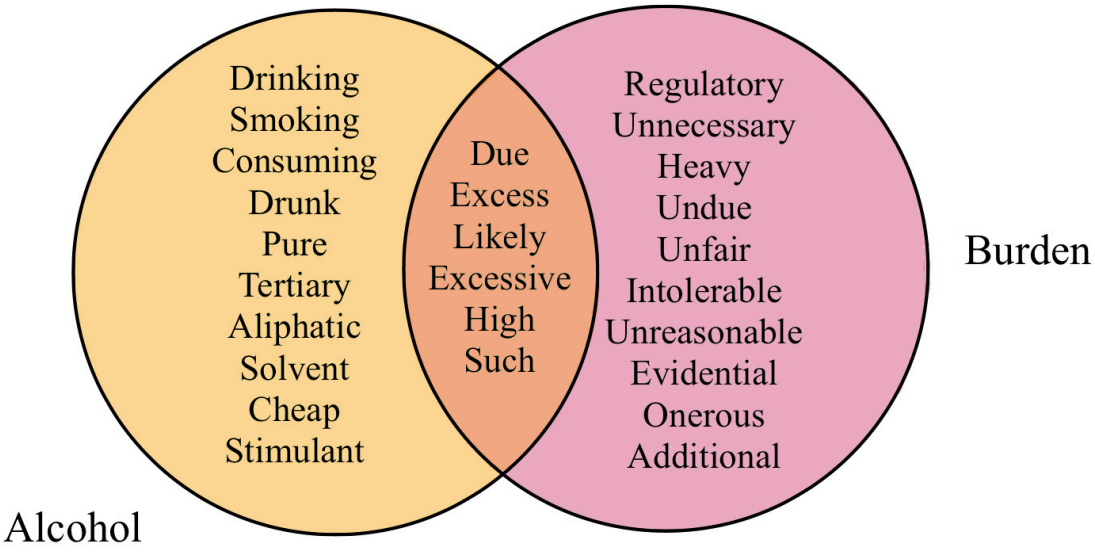

Figure 12. Venn diagram of adjectives retrieved from Sketch Engine (couple alcohol-burden)

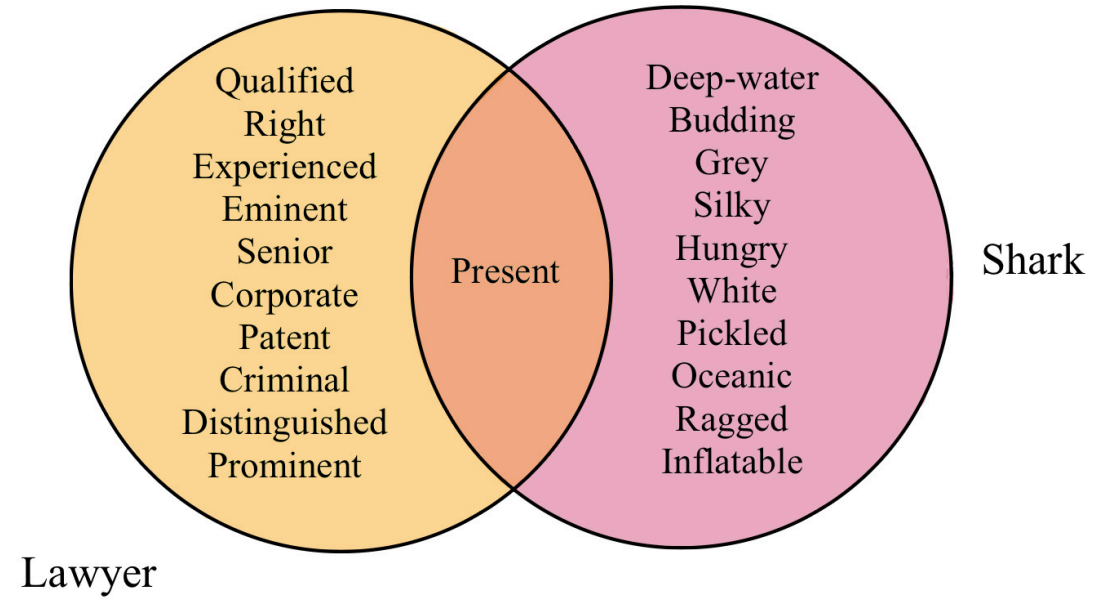

Figure 13. Venn diagram of adjectives retrieved from Sketch Engine (couple lawyer-shark)

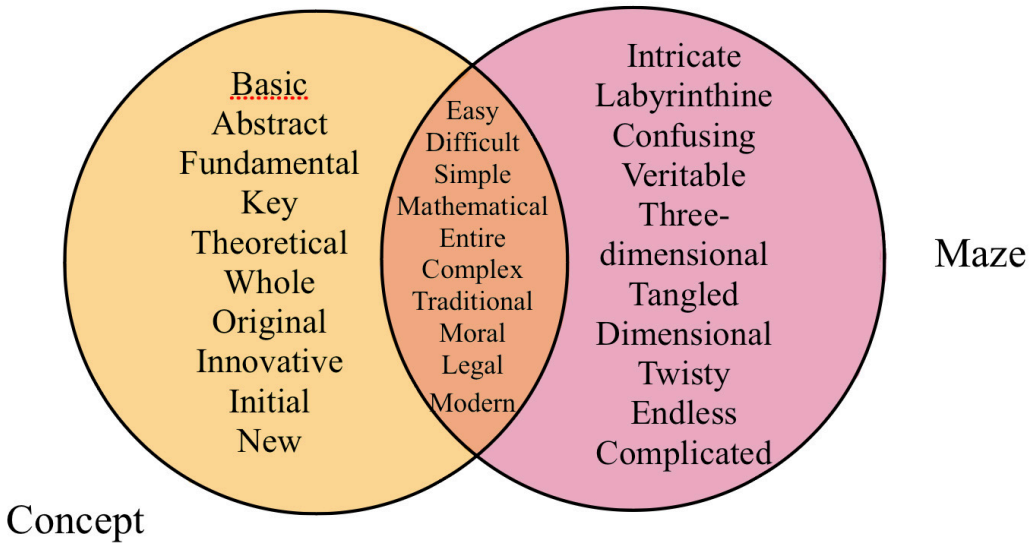

Figure 14. Venn diagram of adjectives retrieved from Sketch Engine (couple concept-maze)

As it is possible to remark from Table 2, only the last pair contains an adjective that may describe the metaphor. However, for *concept – maze* the common list does include other adjectives that not only do not describe the metaphor, but they are also the exact opposite of the 'right' adjective – cf., for instance, 'easy', or 'simple'. It is, therefore, safe to say that the lists retrieved from Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004) returned inconclusive results for experiment 1.

The issue arises with fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016) as well, with the results being most likely worse than those of Sketch Engine.

METAPHORICAL PAIR	ADJECTIVES
Idea – prison	No common elements
Word – razor	No common elements
Party – hurricane	No common elements
Alcohol – burden	No common elements
Lawyer – shark	No common elements
Concept - maze	fiendish, convoluted, labyrinthian, mazy, ziz-zag, tortuous, mazed, Kafkaesque, mind-bending, trackless, intricate, zigzag, labyrinthine, disorienting, circuitous, twisty

Table 3. Adjectives in common between sources' lists and targets' lists retrieved from fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016).

Below, six Venn diagrams will be inserted for each pair with adjectives retrieved from fastText, to visually represent the data inserted in Table 3.

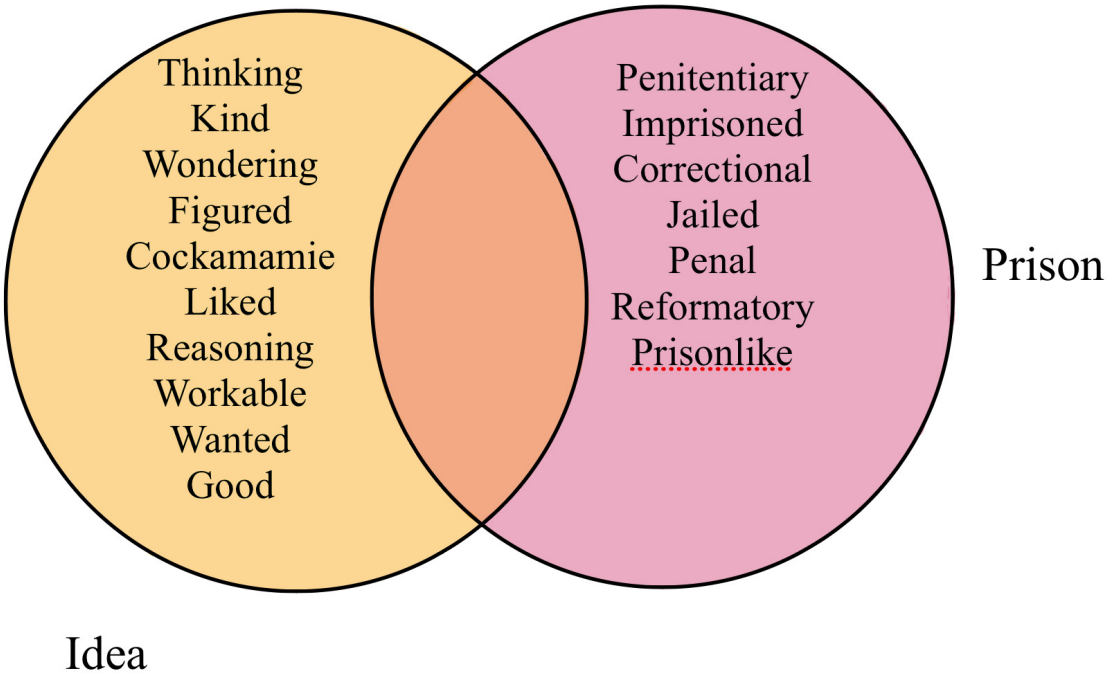

Figure 15. Venn diagram of adjectives retrieved from fastText (couple idea-prison)

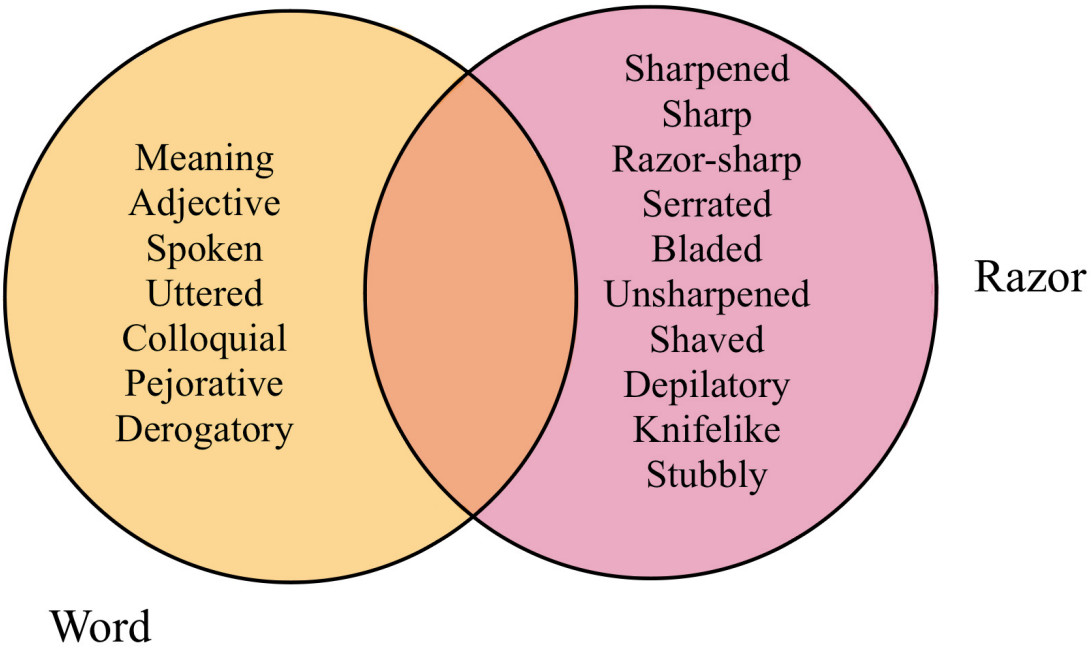

Figure 16. Venn diagram of adjectives retrieved from fastText (couple word-razor)

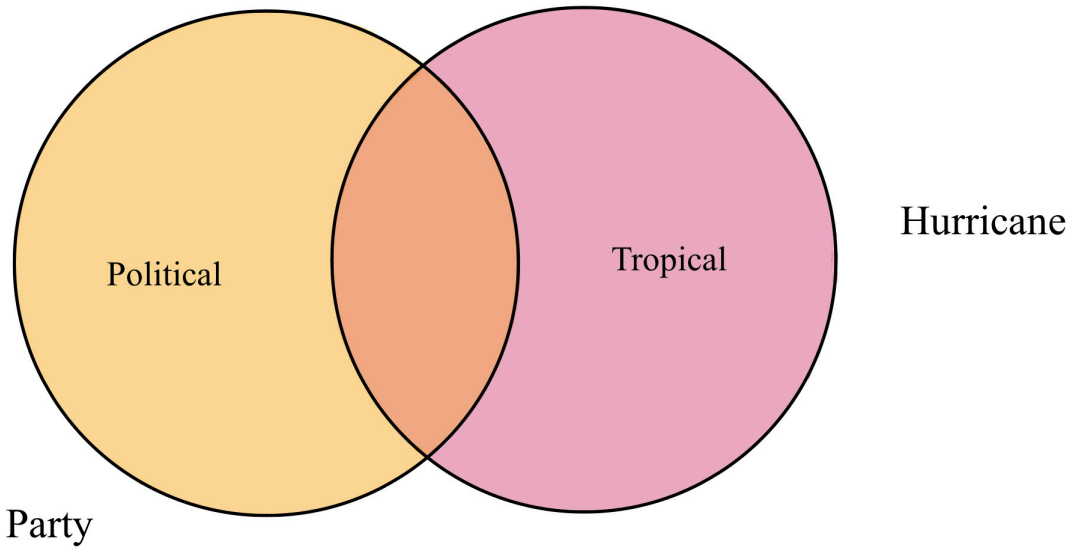

Figure 17. Venn diagram of adjectives retrieved from fastText (couple party-hurricane)

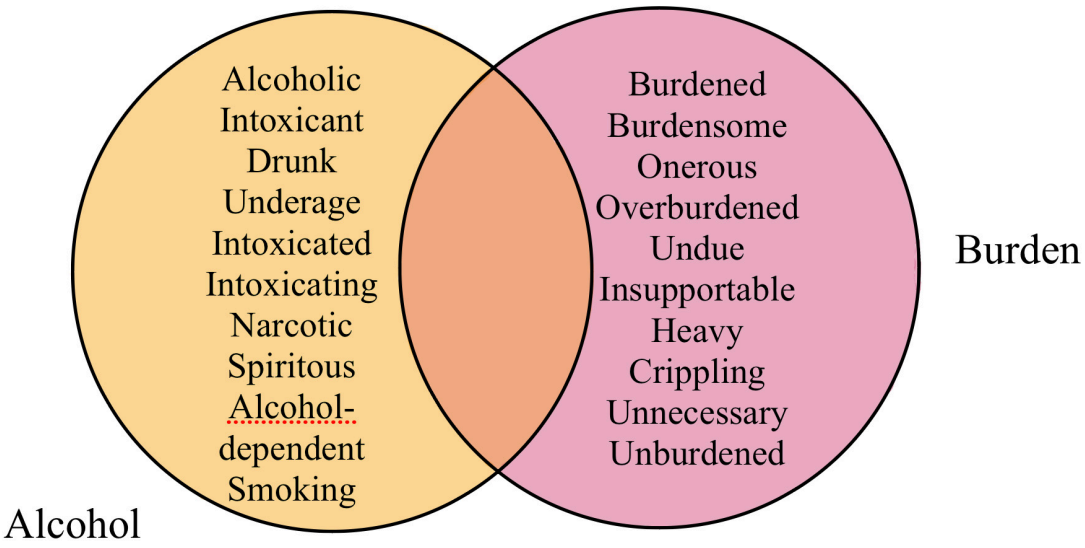

Figure 18. Venn diagram of adjectives retrieved from fastText (couple alcohol-burden)

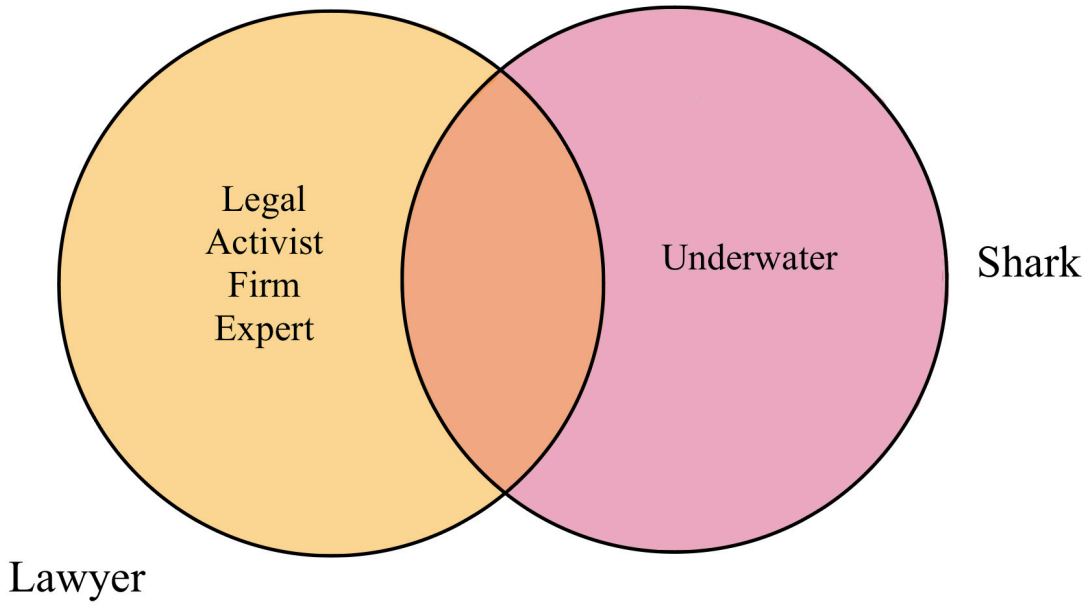

Figure 19. Venn diagram of adjectives retrieved from fastText (couple lawyer-shark)

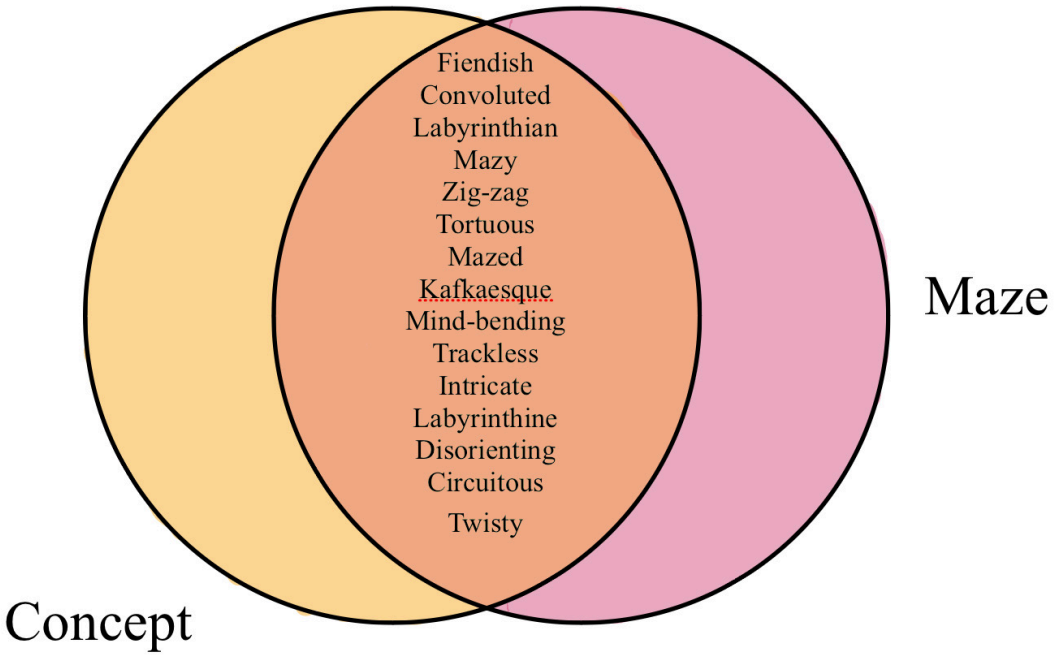

Figure 20. Venn diagram of adjectives retrieved from fastText (couple concept-maze)

As it can be evinced from Table 3, the lists retrieved from fastText for the six randomly chosen pairs never intersected with positive results, except for one case: *concept – maze*. In this case, there are a few adjectives, and therefore not just one, which could describe the given metaphor: e.g., tortuous, mind-bending, intricate, disorienting, twisty. If compared to the list retrieved from Sketch Engine for the same couple, it could be argued that the list retrieved from fastText provides more sophisticated and less general adjectives.

3.4 Experiment 2

3.4.1 Introduction

Experiment 2 derived from the need to adapt the founding idea of experiment 1 to the actual metaphorical mapping.

The issues encountered during the first experiment made it clear that it was necessary to find another computationally cheap approach to achieve the goal; however, the method required to be more complex, in order to best represent the mapping.

The ground for this second experiment comes from behavioural and electrophysiological studies, in addition to Event-Related Potentials studies, on humans, which had the task of understanding how the mapping in metaphors actually works.

The first discover that these studies reached was that processing and comprehending metaphors highly depends on the conventionality of the metaphor itself. This means that by analysing the human response to literal and metaphorical utterances, researchers found longer reaction times and lower accuracy rates for metaphorical expressions compared to literal sentences (McGregor, Agres, Rataj, Purver, & Wiggins, 2019). As it is to be expected, there is also a difference between conventional and novel metaphors: as a matter of fact, conventional metaphors usually require faster reaction times compared to novel metaphors; however, the said reaction times are always longer than those for literal sentences.

Therefore, there is an established proof that the processing – and consequently understanding – of metaphors requires a particular type of mapping, that may be more or less automatic, depending on what metaphors humans are presented with.

It is possible to find different theoretical accounts that tried to best represent the said mapping; nevertheless, the most reliable ones are the *structure mapping model* (Bowdle & Gentner, 2005) and the *career of metaphor model* (Bowdle & Gentner, 2005).

The *structure mapping model* claims that when processing and understanding metaphors, individuals require a symmetric mapping mechanism that allows them to align the commonalities between the source and the target of a given metaphor. In addition, it is also necessary for the mechanism to project an inference about the source onto the target (Bowdle & Gentner, 2005).

The *career of metaphor model* further broadens the previous model by stating that conventional metaphors also need a process of categorisation; instead, novel metaphors do not require such a process, since they are understood in terms of comparison.

These models are highly reliable, backed up by different studies. However, the most serious problem about these models is that they are rightly applied to humans. While it is necessary to understand processes in humans before they are applied to machines, machines and computers do not work in the same way humans do. Therefore, applying these models as they are, intact, it would most likely lead nowhere.

One simple reasoning for this is the fact that the models are based on commonalities between the source and the target. In experiment 1, the attempt was that of using the commonalities and their expected representation in a distributional vector space as a way to choose adjectives that could describe the metaphors. However, as previously demonstrated, this did not reveal itself to be the case. The reason may reside in the fact that machines do not have perceptions.

Bowdle and Gentner (2005) underline how mapping mechanism requires commonalities between the source and the target. Let us say that we have the metaphor in (1), which is said by a kindergarten teacher to her colleagues:

(1) Today my classroom was a zoo

It is true that the characteristics of a zoo are transposed into the classroom she just taught to; however, without the perception of that classroom in that particular day, the teacher would have most likely not even uttered the metaphor. As far as humans are concerned, the perception of the target is a necessary condition for the metaphor to appear. The issue with perceptions is twofold: on one hand, they are personal, so even some other human counterparts may disagree with the metaphor or not understand it; on the other, it is nearly impossible to instantiate a perception in a non-sentient object, like a machine. Computers and machines are not sentient and do not have perceptions, as far as it is known.

Thus, it was imperative to find a way to mimic the mapping without considering the perception role in metaphors. This was reached excluding commonalities represented as adjectives, and trying another approach based on cosine similarity between modified and 'pure' targets (cf. 3.4.2.2 Python's custom-made calc_sim function).

The approach of modelling the transfer of properties from one domain to another has been the key in representing metaphors in computational semantics. Some studies used more structured and logical representations (Martin, 1990), while the most recent approaches are

based on distributional semantics techniques. The reason for this relies on the fact that vector spaces and their diverse features have been demonstrated to improve the performances of models in NLP tasks. Most recently, the method in metaphor interpretation or recognition has grown out of metaphor paraphrases.
While this approach, has been tested several times and has showed to return good accuracies, it does not really represent the 'human' mapping in metaphor understanding. This is why this thesis and this second experiment try otherwise and focus on the specific type of mapping it is possible to observe in individuals.

3.4.2 Methods

3.4.2.1 Materials

Since experiment 2 was expected to employ the correct approach, it was tested on the whole metaphor dataset collected for the thesis (cf. 3.2.1 The metaphor dataset).
Hence, the total number of source-target pair was eighty-four. For each source, its related adjectives were downloaded in a csv file from Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004) and store them in a folder. Most generally, all sources retrieved more than ten adjectives each, apart from four sources (the source is always the second item in the couple, since they respect the *X is Y* order):

- Lie – boomerang: 7 adjectives
- Politician – Chamaeleon: 2 adjectives
- Researcher – bulldozer: 7 adjectives
- Sleep – hug: 4 adjectives

Therefore, in these four cases, the list of adjectives was very short and obviously the function returned all of them, always following a decrescent similarity order.
The adjectives were filtered and stored in a dictionary, which was later called in the function, which will be explained in detail in the next section.

3.4.2.2 Python's custom-made calc_sim function

The theory developed for experiment two, and which was supposed to represent the mapping in human metaphor processing and understanding, needed to be effective but at the same time highly computationally cheap.

The most affordable way to achieve this goal was creating a function on Python. The reason for this being the case is threefold: firstly, the function contains and runs all the steps necessary to achieve the scope with a smooth approach; secondly, it can be inserted in an automatised loop to apply the function – and consequently return the adjectives in this case – for each metaphorical pair; finally, with a bigger model or further pipeline in mind, a Python function is simply accessible, affordable, and on top of everything, useful. Thus, since the idea required some vector manipulation, the function seemed to be the safest place to choose.

At the core of creating the function laid the issue of representing the metaphorical mapping. This was most likely the most significant impediment as far as this first part is concerned.

After a thorough literature review and some extensive brain storming, the choice which made the most sense was that of using adjectives as a mean to interpret the given metaphor. So as to make an example, for the metaphorical pair *time – money*, a possible adjective descripting the metaphor could be 'precious' or 'important'.

Adjectives were the safest choice because they tend to represent certain properties of the noun(s) they are associated to; hence, usually individuals use them when describing a metaphor: most generally, if we are asked to describe what TIME IS MONEY means, we would probably say 'time is precious' or 'time is non-infinite'. Basically, speakers stripe away the metaphorical association and assign an adjective (deriving from the source) to the target.

Starting from this viewpoint, the goal was that of using this diffused process and apply it to a function.

Since the processes inside the function – called *calc_sim* – are quite diverse and difficult, these will be broken down in the following few lines, in order to make them as clear and easy as possible.

The first step in building experiment two's function was that of retrieving the vectors since the whole concept is based upon vector manipulation. In particular, the vectors retrieved were those for each adjective referring to the source of a given pair, plus the vector for the pair's target. As previously stated, the model used for the vector retrieval was fastText (Bojanowski,

Grave, Joulin, & Mikolov, 2016) (cf. 2.3.2. The role of context, distributional spaces, and fastText).

Once the function had retrieved all the necessary vectors for each adjective and the target, the adjectives were stored in a list, which would maintain the order of original list of adjectives – the only difference was that, in this last case, it was a list of vectors and not a list of strings.

The second step consisted in manipulating the vectors. As clearly explained in the introduction of this experiment two's section, the scope was that of modifying the target, as to represent the mapping in metaphor processing. In order to do that, two simple mathematical operations were adopted: multiplication (i.e., Hadamard product) and addition.

Therefore, the function took every adjective vector in the list and multiplied it for the target vector. The same process was carried out a second time, where, however, the vectors were not multiplied but one was added to the other.

As a result, the function created a new series of modified vectors which should represent the target in a given pair, but with a different shade of meaning to it, represented by the adjective vector. These resulting vectors were again inserted in another list, so as to respect the order in which the adjectives originally appeared.

The third step comprises the calculation of cosine similarity. The rationale deals with the fact that for the theory to work, the 'correct' adjective (which should interpret the metaphor) is an adjective that does modify the target, but not too much as to change its meaning. Hence, in order to make this idea computational, cosine similarity was the approach that best suited the search for said adjective.

Cosine similarity is one of the most used measures in DS to represent the degree of similarity between two nouns. More specifically, it identifies a number – always between -1 (not similar at all) and 1 (completely similar) – that corresponds to the cosine of the angle which two vectors form when it is required to investigate how similar they are. The reason for this is that in distributional spaces, words that have similar meaning will have similar distributions. Therefore, if we calculate the similarity between the vector for 'car' and the vector for 'automobile', the value of cosine similarity will be closer to 1 compared to the value of cosine similarity calculated between the vector for 'car' and the vector for 'palm'.

Thus, the function calculates the cosine similarity between each modified target and its original 'pure' target for each source-target pair. Next, the similarities are inserted in another list to respect the order.

The fourth and final step directs the process to its most delicate part: sorting and identifying the acceptable results. The function was required to return the highest similarities, since it

would mean that the adjectives related to those similarities were modifying the least their target. Therefore, sorting the lists both for Hadamard product and addition was the easiest solution. Once the lists were sorted, a simple splitting was carried out, in order to consider only the first ten elements in the sorted lists. After this splitting, the problem was retrieving the adjectives corresponding to those first ten similarities.

Firstly, a for loop was inserted in order to calculate the index of each similarity in the non-sorted original lists. These indexes were used to then retrieve the adjectives from the adjective lists. As it can be understood, this process was executed both for Hadamard product and addition; therefore, in total, the function would return and print out two lists of ten adjectives each, plus the corresponding similarities.

```
free-flying: [[0.04676428]]
purple: [[-0.00419345]]
brown: [[-0.00493237]]
active: [[-0.01344571]]
common: [[-0.01409401]]
such: [[-0.03061228]]
tropical: [[-0.03220157]]
white: [[-0.03440734]]
orange: [[-0.04640913]]
plentiful: [[-0.04688353]]
```

Figure 21. Results of the ten adjectives with highest similarity for Hadamard product (acrobat-butterfly)

```
beautiful: [[0.91340804]]
colourful: [[0.8934239]]
present: [[0.8892555]]
free-flying: [[0.8820038]]
endangered: [[0.8770472]]
numerous: [[0.8746999]]
iridescent: [[0.8740717]]
plentiful: [[0.871848]]
likely: [[0.8248985]]
common: [[0.8178724]]
```

Figure 22. Results of the ten adjectives with highest similarity for addition (acrobat-butterfly)

Figure 15 and Figure 16 show what the function printed for the couple acrobat – butterfly. The order of the adjective respects the similarity value (from high to low) and, more importantly, the list of total adjectives has been reduced to the first ten adjectives, which gives the machine more opportunities to achieve the goal and may return more than one appropriate adjective. The numbers next to the adjective are the similarity values between the 'pure' target and the target modified with the given adjective.

3.4.3 Initial results

Before conducting a more scientific analysis and, more specifically, asking humans to rate the precision of the machine, an informal check of the data was carried out.
Since for the survey the possible adjectives to insert in the list needed to be either the result of Hadamard product or addition, and not both, I 'rated' if the machine was able to return more appropriate adjectives through Hadamard product or addition.

From this informal preliminary analysis, it became clear that for the majority of the metaphorical pairs, the adjectives were the same for both addition and Hadamard product. However, addition failed at returning at least one good adjective for seven pairs; while for fourteen out of 84 pairs, neither Hadamard product nor addition found an appropriate adjective. It is extremely important to underline that this is in no way an official analysis: it was necessary however to choose from a human perspective which of the two mathematical operations returned the highest number of correct adjectives. Since the rules were strict and the metaphors were highly conventional, it was possible to carry it out without difficulties and without doubts.

For a more scientific and official analysis of the machine's precision in returning at least one adjective per metaphorical pair, please refer to the next section, where the survey distributed to human judgers will be explained in detail.

3.4.4 Qualtrics' survey

Once the function had been run for all eighty-four metaphors, and the results were collected, it was necessary to find a way to analyse them with a scientifically reliable approach. The most suitable method was that of creating a survey that would ask human judges to define whether for each metaphor the machine had returned at least one appropriate adjective.

The reason for this being scientific is that individuals have been proved to be able to understand and describe conventional metaphors extremely well – although the reaction times may be slower compared to literal sentences (McGregor, Agres, Rataj, Purver, & Wiggins, 2019). Therefore, having an adequate number of speakers react to and ratify the precision of the machine in interpreting linguistic features it does not understand would allow to obtain a reliable measure of accuracy.

In order to receive individuals' choices, it was necessary to create and distribute a survey. For this thesis we relied on Qualtrics[4],.

The survey started by showing three separate pages with instructions on what the survey consisted of, how to carry it out (e.g., only in one session and through a computer), the module of consent explaining the right to withdraw and not accept to do the survey, and the actual question of acceptance.

A fourth introductory page was required in order to exclude participants whose English language level was not sufficiently high. Indeed, participants were required to possess – either through an official certification or from a general guess based on university exams – at least a B2 level in the CEFR (Cambridge Assessment English, n.d.). Hence participants were presented a question where they had to indicate their level of English according to the already mentioned CEFR while being as sincere as possible. They had six options, corresponding to the levels: A1, A2, B1, B2, C1, C2. Three *skip to* functions were inserted in case participants selected one of the first three levels (namely, A1, A2, B1). If this happened, the software directly skipped to the last page, since they obviously did not have the level to be able to process metaphors in English as they should have.

This fourth page, and first real question, was by no means intended to discredit or judge the participants. However, it was necessary for the individuals to process the whole survey in

[4] The survey was conducted using Qualtrics survey, version [March 2022]. Copyright © 2022 Qualtrics. Qualtrics and all other Qualtrics product or service names are registered trademarks or trademarks of Qualtrics, Provo, UT, USA. https://www.qualtrics.com

English, think of adjectives that could describe the given metaphor and decide whether those listed were appropriate. If participants did not have a level high enough, it would have not been possible to consider their answer as reliable.

As it can be foreseen, all metaphorical pairs needed to be presented an equal number of times to the participants. Nevertheless, creating a survey with eighty-four questions and forcing participants to make a decision for all of them would have been an overload. Therefore, the dataset was randomly split into three groups, each containing twenty-eight pairs. The pairs were not randomly assigned to the group since there was no pattern in the dataset: the first third of the dataset constituted Group 1; the second third was inserted in Group 2; and the remaining pairs formed Group 3.

For each pair, participants would see the prompt "Choose at least one adjective in the following list that you think may interpret the meaning of this sentence: *X is Y*", where *X is Y* constituted the relationship between the source and the target of each pair. In certain cases, to make the questions more diverse, a shorter prompt was added, e.g., "Look!" or "I can't believe that". These adjuncts did not alter the metaphor, since the relationship was always of the type X is Y, simply they avoided that the survey's questioned felt repetitive. Figure 17 gives an example of the question showed to participants for the first conceptual metaphor of the dataset: AN ACROBAT IS A BUTTERFLY.

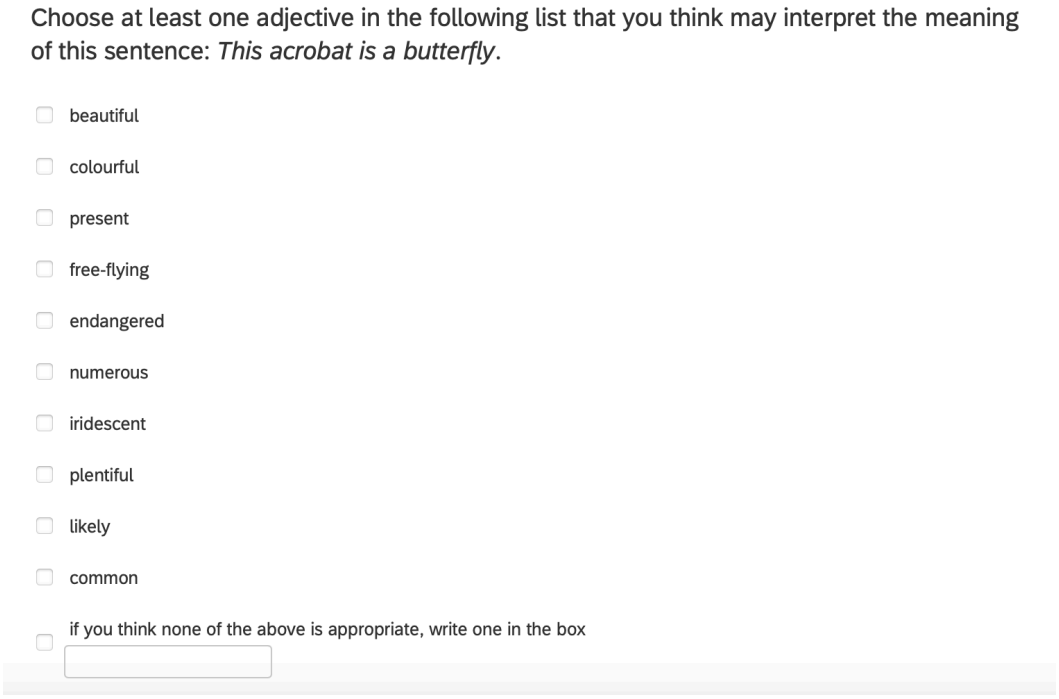

Figure 23. Example of question on Qualtrics given to participants (couple: acrobat-butterfly)

Each question contained eleven choices. The first ten choices were the adjectives returned through the Python's function *calc_sim*. They were inserted for visual purposes in the order given by the machine; however, once the survey was distributed to the participants, these would see the choices in a randomised order.

The eleventh choice contained a standard prompt "If you think none of the above is appropriate, write one in the box". This choice could be selected, however participants had to insert at least one adjective they thought was more appropriate. In addition, it is important to underline that all questions allowed multiple answers. Hence, if participants thought one of the ten adjectives could describe the metaphor but also thought of another appropriate adjective, they could certainly do so.

Once the participants clicked on the survey link shared with them, they were randomly assigned a group of questions.

Each participant answered to twenty-eight questions, and once they completed the survey, a final page with an inscription thanking them for completing the survey and notifying them that their answer had been stored.

The survey link was shared among university students, through WhatsApp groups, Facebook groups and personal acquaintances. After two months the number of people completing the survey plateaued at zero, and there was no way of collecting more replies. Therefore, the survey was ended there, with a total of fifty-four participants, i.e., eighteen participants per group.

3.4.5 Final results

The total number of total responses was fifty-four, while the total number of analysable responses recorded by Qualtrics was fifty-one. Three responses could not be considered because those participants did not have an English level of B2 or higher, and therefore they were not even able to answer to the following questions.

To give a description of the English levels of the fifty-four participants, 1 participant declared to have an A2 level, 2 participants stated they possessed an B1, 20 claimed they had a B2, 25 individuals declared their level was C1, and finally, only 6 affirmed they possessed a C2 level.

The data collected through human judges was analysed according to the rules of precision-at-k. *Precision* is together with *recall* one of the most used techniques to calculate accuracy and effectiveness in information retrieval (Manning, Raghavan, & Schütze, 2008, p. 154). Precision is calculated as the fraction of retrieved items which are relevant to a certain task. More practically (Manning, Raghavan, & Schütze, 2008, p. 155), it would be:

$$Precision\ (P) = \frac{number\ of\ retrieved\ relevant\ items}{number\ of\ retrieved\ items} = P(relevant|retrieved)$$

Considering the task for this thesis, the 'number of retrieved relevant items' would correspond to the number of adjectives that could describe the metaphor according to human judges; instead, the 'number of retrieved items' would be equal to 10 (sometimes less then 10 because the list of adjectives was very short, see 3.4.2.1 Materials).

The most useful and most reliable precision would be the one considering all adjectives, therefore precision-at-10. This is the type of precision that was be chosen to carry out an evaluation of the approach. However, more than one precision was taken into consideration. Indeed, cumulative precision at every adjective was calculated, for a total of ten values: precision-at-1, precision-at-2, precision-at-3, etc.

A relevant question was that of when it would be sensible to assign 1 and when 0. In other words, in order to effectively calculate the precision, the adjectives needed to be assigned a number – 1 meant that it received some votes, 0 meant that no participant chose that particular adjective. It felt necessary to conduct two analyses: the first analysis has a 'threshold' at 1, while the second analysis has a 'threshold' at 2. This means that in the first case, every adjective which had at least 1 vote from the human judges would have been assigned value 1, otherwise

it was assigned the value 0. In the second case, instead, only adjectives which received at least 2 votes were assigned value 1.

After all the partial precisions were calculated (namely, the machine's precision at each single question or metaphorical pair), the mean average was computed for each precision for two reasons: first, have a grasp about the general accuracy at each *k*; second, be able to plot a graph where the trend would have been made visible. The mean averages and the graphs were calculated and plotted for each 'threshold'.

The final results and the graphs will be inserted below, before continuing with a more careful analysis and consideration – precision-at-*k* will be shortened to 'p@*k*'.

p@1	p@2	p@3	p@4	p@5	p@6	p@7	p@8	p@9	p@10
0,726	0,702	0,677	0,659	0,637	0,623	0,625	0,627	0,629	0,611

Table 4. Mean averages of single precisions for each k. Threshold at 1.

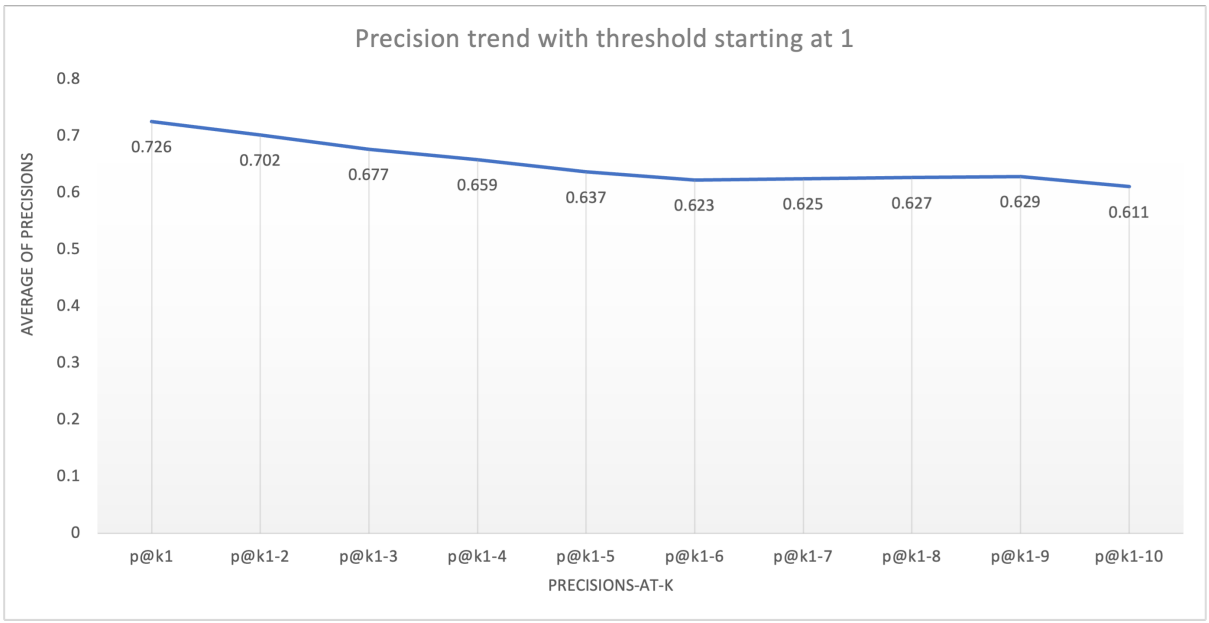

Figure 24. Graph of the trend of averages for each precision (threshold = 1). Averages are signalled as labels below the trendline.

The first analysis conducted assigning to each choice a value of 1 if it had received at least one vote returned averagely significantly relevant results. As it can be seen from Figure 24., the best precision was returned for precision-at-1, with a value of 0.726. The problem of

considering only the first result as appropriate may rule out some other appropriate adjectives which may describe the metaphor. In certain metaphorical pairs, the first adjectives received at least 1 vote, however an adjective with a lower index (i.e., closer to 10) received significantly more votes – to give a demonstration, for the pair acrobat-butterfly adjective #1 received 7 votes, while adjective #4 received 16 votes.

The trend's slope always report a difference of about 0.20 points until precision-at-6. Subsequently, the slope does raise – albeit not as significantly as the decrease – between precision-at-6 and precision-at-9, to simply decrease again at precision-at-10.

As far as the second analysis is concerned – i.e., where only adjectives with at least two votes were assigned value 1 – the results were slightly less performant, as it can be imagined. Before continuing with the analysis, Table 5 and Figure 25. show the data derived from the above-mentioned analysis.

p@1	p@2	p@3	p@4	p@5	p@6	p@7	p@8	p@9	p@10
0,643	0,554	0,540	0,531	0,499	0,480	0,465	0,470	0,463	0,459

Table 5. Mean averages of single precisions for each k. Threshold at 2.

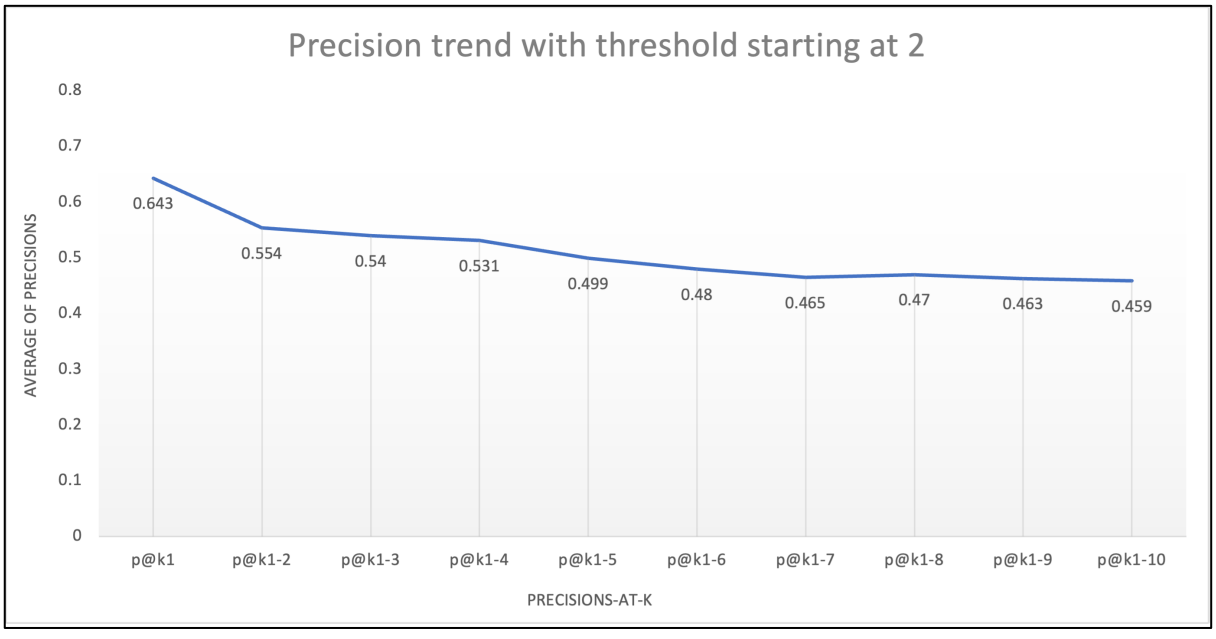

Figure 25. Graph of the trend of averages for each precision (threshold = 2). Averages are signalled as labels below the trendline

Since a smaller number of adjectives was held appropriate in this second analysis, all precisions are averagely 0.1 points lower compared to the first analysis. This is not surprising, nor it was expected any different. However, there appears to be a different trend, if compared to the first analysis. Between precision-at-1 and precision-at-6 there is a downward slope; however, the pace of decrease is not as steady and consistent as the one in the first analysis (in the same range of precisions): here, the slope does not decrease with an overall homogenous reduction. It is also possible to remark that there is no slight increase between precision-at-6 and precision-at-9: only a minor increment of 0.02 at precision-at-8 can be observed; nevertheless, it does not follow the same trend which can be found in the previous analysis.

Overall, it is safe to assert that the two slopes identifying the trends for the two analyses are not dramatically different: indeed, both show a trend's downturn after precision-at-1, although less steep in the first analysis. They also both demonstrate that the precision-at-10 – which is argued to be the most informative as previously delineated – is significantly lower compared to precision-at-1, although this is less evident for the first case, compared to the second analysis. This last remark is in line with the fact that in the second analysis more adjectives have been ruled out, and therefore, it is in fact obvious that the difference between precision-at-1 and precision-at-10 is more significant in the second case. Figure 26. demonstrate in a visual way the reasoning that has been carried out in the previous lines.

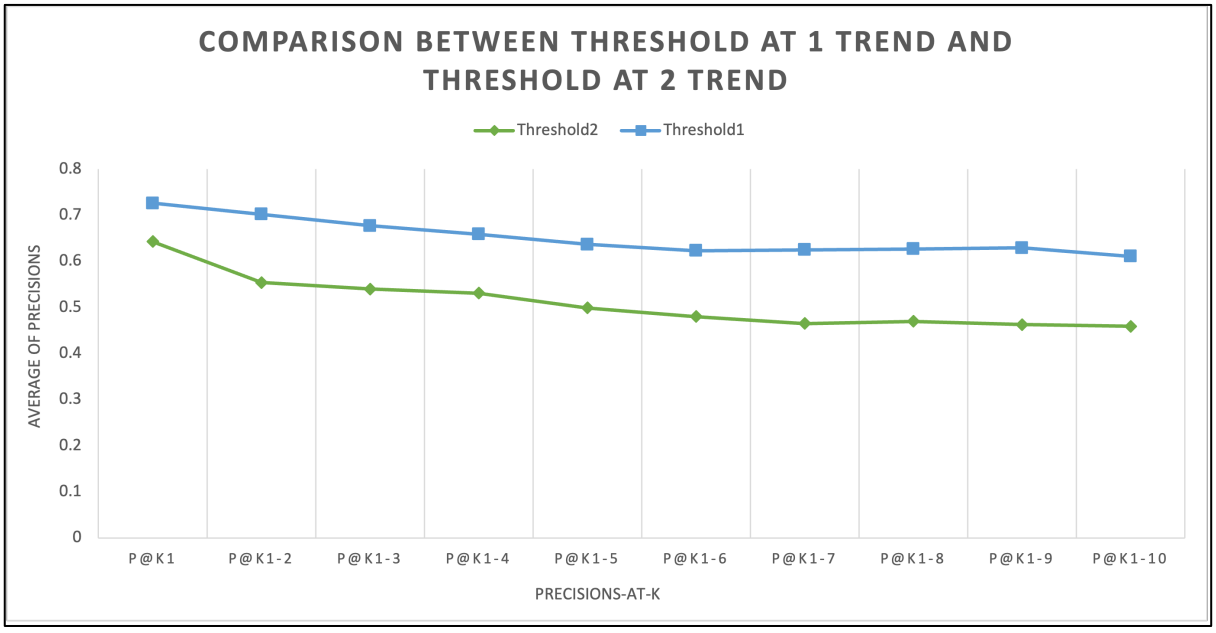

Figure 26. Comparison between the trends identified in the two previous graphs.

In conclusion, when considering all adjectives which received at least one vote by human judges, the machine (i.e., the function) does seem to perform significantly better. When ruling out the adjectives with less than two votes, the performance does deteriorate to a certain extent, although it would be unfair to claim that the results are catastrophic.

The first two analyses were necessary to have a general understanding of how the machine performed and what could be done otherwise. However, from a linguistic point of view, it is informative to a certain extent.

Since Coli (2016) reports psycholinguistic ratings, for instance *familiarity* and *innovativeness* (cf. 3.2.1 The metaphor dataset), it would have been interesting to investigate what relationship – if any – exists among the precision of the machines and the said features. The most sensible way to carry out this task was by performing a multiple regression between four variables: the dependent variable was the precision-at-10, which was said to be the most reliable although lower compared to precision-at-1; instead, the independent variables were selected among the features by Coli (2016), in particular *familiarity*, *innovativeness*, and *comprehensibility*. It is important to highlight, however, that this third analysis employing multiple regression was conducted only on seventy-four out of eighty-four source-target pairs, which are present in the dataset collected by Coli (2016). The last ten metaphors which had been added for the image classification task did not undergo the same judging process as the other metaphors, and were, therefore, excluded from the multiple regression.

The three features chosen were not picked randomly, instead, they followed a reasoning. The judgements collected by Coli (2016) for her thesis were based upon characteristics that normally non-literal language has. Among these, three – those mentioned a few lines above – were particularly interesting, because they may be either directly or in directly represented in a corpus (used to retrieve the vectors).

For instance, if a metaphor is familiar or highly comprehensible for humans, it essentially means that individuals have encountered it many times, their processing reaction times are potentially shorter. From a computational perspective, if a word or phrase is familiar to people, it means it occurs frequently in texts, discourses or conversations which are later used to build corpora.

Innovativeness displays an opposite behaviour compared to *familiarity* or *comprehensibility*, since if a metaphor is defined innovative, it means it is new – albeit not novel – and not as sedimented as highly conventional metaphors. As a consequence, if a metaphor is innovative

for humans, it is fair to expect that a corpus will not display it as often as familiar or comprehensible metaphors.

This reasoning has some implications for the machine's precision. In theory, when the function has to return the adjectives for a familiar metaphor, it would be reasonable to expect a better performance than one returning adjectives for an innovative metaphor. The reason for that being the case is that the target and the source appear more frequently together in sentences: this leads to more appropriate adjectives in the source's list and more significant chance of having a higher cosine similarity for the said appropriate adjectives. Thus, the value for precision-at-10 in this case – which consider the totality of adjectives returned – should be higher compared to the value of precision-at-10 for innovative metaphors.

As previously stated, a multiple regression has been conducted to test whether the reasoning was supported by the machine's results or if it is faulty.

The most important value, which does offer an insight on the relationship among these variables is R^2, which was not significant since its value attested at 0.1. If R^2 has a value of 0.1, it is widely accepted that the model does not explain more variability; hence, it can be asserted that there is not a significant effect, since the value of R^2 is this low.

There is, notwithstanding this, an interesting pattern as far as innovativeness is concerned, which does seem to counter the argumentation made before about the expectation as far as *innovativeness* and *familiarity* or *comprehensibility* are concerned.

Model		Unstandardized	Standard Error	Standardized	t	p
H_0	(Intercept)	0.625	0.020		30.562	< .001
H_1	(Intercept)	0.231	0.164		1.411	0.163
	comprehensibility	0.038	0.024	0.207	1.597	0.115
	familiarity	0.013	0.012	0.115	1.005	0.318
	innovativeness	0.038	0.016	0.314	2.434	0.018

Table 6. Summary of coefficients' calculation. The most interesting value to remark is the p-value for innovativeness.

By looking at Table 6, only *innovativeness* possesses a p-value below 0.05, since its value is 0.018. This value casts light on the possibility that the more innovative the given metaphor is, the more likely it is that the machine will demonstrate a high precision in its interpretation. On

the other hand, *comprehensibility* returned a p-value of 0.115, while *familiarity* had the highest p-value, namely 0.318.

However, there could have been some interferences of multicollinearity; hence, to rule out the possibility that there is an effect of multicollinearity, the variance inflation factor (VIF) has been calculated. The VIF factor, nevertheless, did not show any sign of multicollinearity, since its value was below 5, leading to the conclusion that there was no evidence of multicollinearity between *innovativeness* and the other variables.

A further analysis on the behaviour of *innovativeness* has been completed, by plotting the marginal effects for the said feature. Marginal effects are interesting because they display the effect of a certain variable – in this case *innovativeness* – while the other variables are kept stable.

Figure 27 shows the plotting of the before-mentioned marginal effects, where the 95% confidence interval has been inserted and displayed by the grey area around the black trendline.

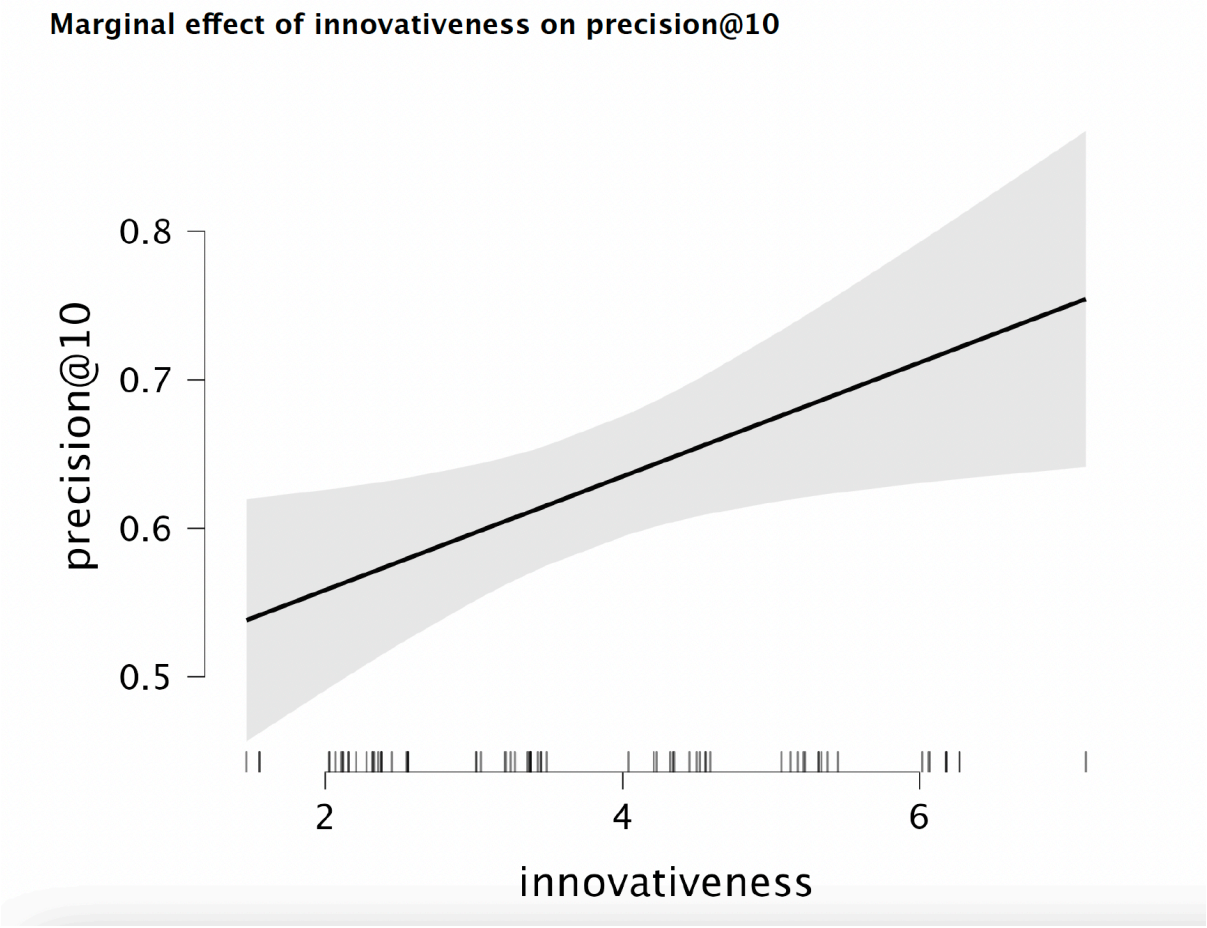

Figure 27. Plot of the marginal effects of innovativeness on precision-at-10. The grey area around the trendline represents the 95% confidence interval.

The case of innovativeness is interesting because it could lead to significantly positive results with novel metaphor interpretation, and this would facilitate the treatment of said metaphors.

3.5 Image classification task

3.5.1 Introduction

Although a well-made, thought-through, and reliable corpus of visual metaphors does not exist for the time being (cf. 3.1 The background), the original purpose of the thesis was that of creating an approach that could process and interpret visual metaphors.
Hence, as explained a few sections before (cf. 3.1 The background), the first step of recognising the source and the target in a given visual metaphor was skipped and set aside for a future moment when a reliable corpus will be available. Nonetheless, even though the first step was not included in this project, it was still necessary to carry out an image classification task.
The reason for this being the case is that before the machine has the possibility of using the *calc_sim* function and actually return some adjectives that could describe the metaphor contained, it requires to have the string – i.e, the name – of both the source and the target. This is possible exclusively through image classification, for the machine needs to be able to recognise what is represented in the picture.

Since a pipeline was felt inevitable, it was imperative to carry out an image classification task before proceeding with the said pipeline. However, this thesis was not meant to gravitate around *Computer Vision*; hence, the task was not required to discover or test some particularly innovative techniques or some specific and highly detailed dataset. The scope was that of managing to assemble a pipeline for visual metaphor interpretation, and therefore an image classification task was necessary in order to obtain the correct classes of the images representing sources and targets. Thus, both the methods and the Neural Network (NN) used are coherent with the current state-of-the-art.
It is worth mentioning, however, the few issues encountered in this section of the project. The first problem derived from retrieving the images. As it will be addressed in the next section, the databased chosen was ImageNet (ImageNet, 2021), because of its vast availability of classes and images. It is not an uncommon practice in image classification tasks, to use a subpart of the whole 14-million-image dataset, consisting of 1,000 classes for a total of 1,281,167 training images, 50,000 validating images and 100,000 testing images (Russakovsky, et al., 2015). However, in this dataset, the classes did not contain the classes corresponding to the sources and targets in Coli's (2016) dataset. Therefore, the only solution was that of asking for permission and downloading the whole dataset.

Nevertheless, although permission was granted and the entire dataset was downloaded, retrieving the images for all sources and targets of the Coli's (2016) dataset was not possible: many of the sources and/or targets were not represented with images on ImageNet.

For the aforementioned motivations, the current work is a proof of concept. The experimental setting has been conducted under a constrained environment, in order to obtain some initial, baseline results. Bear in mind, though, that image classification is already established to give superb results (Russakovsky, et al., 2015), and it is used in all various settings. Hence, this section needs to be laid out as a proof of concept, for coherence purposes; yet, even though more pictures are retrieved, the results would not most likely drastically change.

3.5.2 Methods

As previously explained, only a small part of the dataset used for the NLP side of the project had been represented with images in ImageNet and was, therefore, available to be used. In particular, the CV task could afford to employ only 20 source-target pairs. It is possible to find the list in the table below.

Acrobat-butterfly	Man-chest	Professor-rock	Cloud-cotton
Dancer-drangonfly	Man-sewer	Flower-blanket	Garden-rug
Giraffe-skyscraper	Professor-machine	Snow-blanket	Home-prison
Girl-armor	Soldier-lion	Man-lion	Heart-rock
Girl-flower	Street-snake	Dancer-swan	Time-money

Table 7. List of source-target pairs used for the image classification task.

It is important to make a note: for some of the specific sources or targets, still there were no archives in ImageNet. However, by using synsets, it was remarked that there were archives for less specific nouns. For example, there were not tars for 'girl' and 'man'; however, there was a tar for 'person'. Now as far as the representation, this is not a major issue, since a girl and a man are biologically people. Yet, it does impact the results in a pipeline. Thus, a condition will

need to be made available (cf. 3.6 The pipeline). Through NLTK (Loper & Bird, 2002), it was possible to retrieve the tars of the corresponding synsets linked to both sources and targets (cf. 3.2.3 Using WordNet).

The images and their corresponding labels were split into three sets – a train set, a validation set, and a test set. The train set contained 32,267 images, the validation set contained 4,019 images, and the test set contained 4,058 images.

The framework for the entire image classification task is the one suggested by PyTorch (Paszke, et al., 2019): every step was carried out according to the said framework.

Before training and testing the NN for the classification task, a custom dataset was built, so that the datasets respected the splitting previously made. With a function, the images from the tars underwent a resize (224x224) and were assigned the corresponding labels, which had been previously stored in a csv file. There was an instance where a picture did not have the standard RGB channels for coloured picture but was in black and grey. Therefore, the function also had the goal of repeating the black-and-grey channel, 3 times, in order not to return any error while running the code.

As far as the NN is concerned, the choice fell on ResNet (He, Zhang, Ren, & Sun, 2015) for two reasons: first of all, it constitutes a significantly and widely used model in CV, which delivers a fair sense of reliability; secondly, ResNet has different versions, each containing a different number of hidden layers, and this reduces the risk of overfitting. If the structure of the NN does not match the difficulty of the task, there is the possibility of running up against the NN learning the noise, or learning all the patterns, which would lead to an eventual underfitting. For this project, ResNet50 was employed, given the fairly small number of images in the dataset. NNs have a set number of hidden layers – which are responsible for training and making predictions – which are proper to each individual NN. In the case of ResNet50, the number of layers is represented by the number in the NN's name, namely it has 50 hidden layers. In the case of ResNet18, the NN has 18 hidden layers, and so forth.

The training and the testing of ResNet18 were conducted with the remote machine's GPU, with a reproducibility seed set at 42[5]. Finally, the training was iterated for 1,000 epochs, with a batch size of 16 and a learning rate of 0.001.

[5] The reason can be found in *The Hitchhiker's Guide to the Galaxy* (Adams, 1979)

3.5.3 Results

The results of the image classification task refer to the three phases normally used to train and test the NN: training phase, validation phase, and testing phase. The analysis of said results will follow the order just given and will take into consideration two values which were calculated by the NN at each epoch, namely the *loss* and the *accuracy*. The loss represents the value assumed by the loss function – in this case, Cross Entropy[6] – at each iteration, and interprets the inaccuracy of the NN's predictions. The accuracy was simply the value corresponding to: $\frac{correct\ predictions}{number\ of\ observations}$.

As far as the train set is concerned, both the loss and the accuracy follow an expected trend: the higher the epoch, the lower the loss value and the higher the accuracy. This was to be expected simply because while the NN is technically learning to make predictions based on the dataset, a high number of epochs should lead the NN to perform better, unless it is entangled in an overfitting or underfitting situation. Since every choice was pondered on the type of dataset, the type of task and the desired outcome, it was fair to await good results. Before continuing, here are the graphs representing loss's and accuracy's slopes.

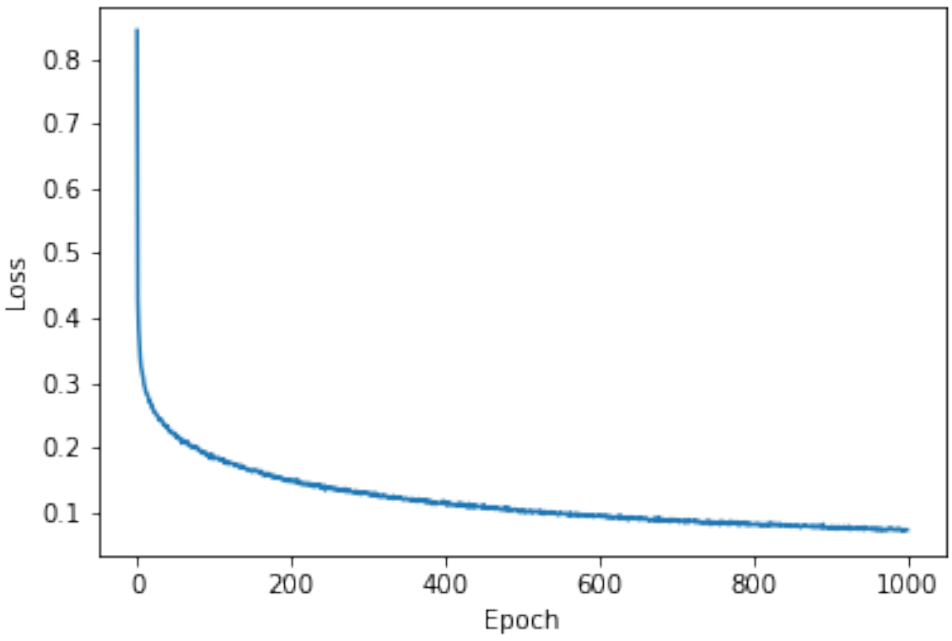

Figure 28. Curve of the NN's loss values over 1000 epochs (training set)

[6] For a detailed description of Cross Entropy, please refer to pytorch.org

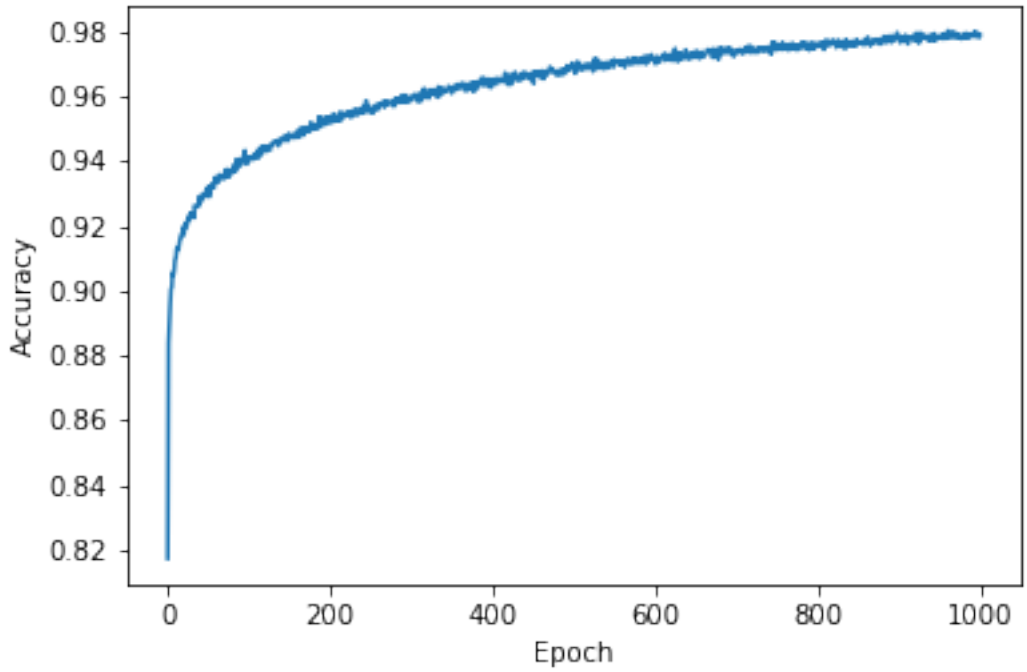

Figure 29. Curve of NN's accuracy values over 1000 epochs (training set)

The plots do show significant and reliable results for both loss and accuracy values. Towards the end of the training, the loss value was close to 0.1, while the accuracy was almost 0.98 (i.e., 98%). The graph shows minor and small fluctuation in the slope over epochs: this means that the increase of accuracy or the decrease of the loss were not as steady as it may be expected. This is considered to be quite normal, and it may depend on the NN or the dataset. The most important event that it is desired to be measured is that the loss overall tends to decrease, and the accuracy overall tends to increase. This is the case for the training phase of this dataset, and therefore it can be claimed that the model capabilities are adequate to learn the classification task and in line with previous image classification tasks.

As far as the validation phase is concerned, the situation is different compared to the previous phase. More generally, the validation phase is supposed to give an estimate of the skill and robustness in making prediction of the previously trained and model (which has been carried out in the training phase). The validation phase is extremely important because this estimate constitutes the base upon which the decision of which model is important to save. In other words, at every epoch the training algorithm is instructed to calculate if the model accuracy on

the new epoch is higher compared to the previous epoch and, if so, save the weights of the best model so that it can be used in the testing phase.

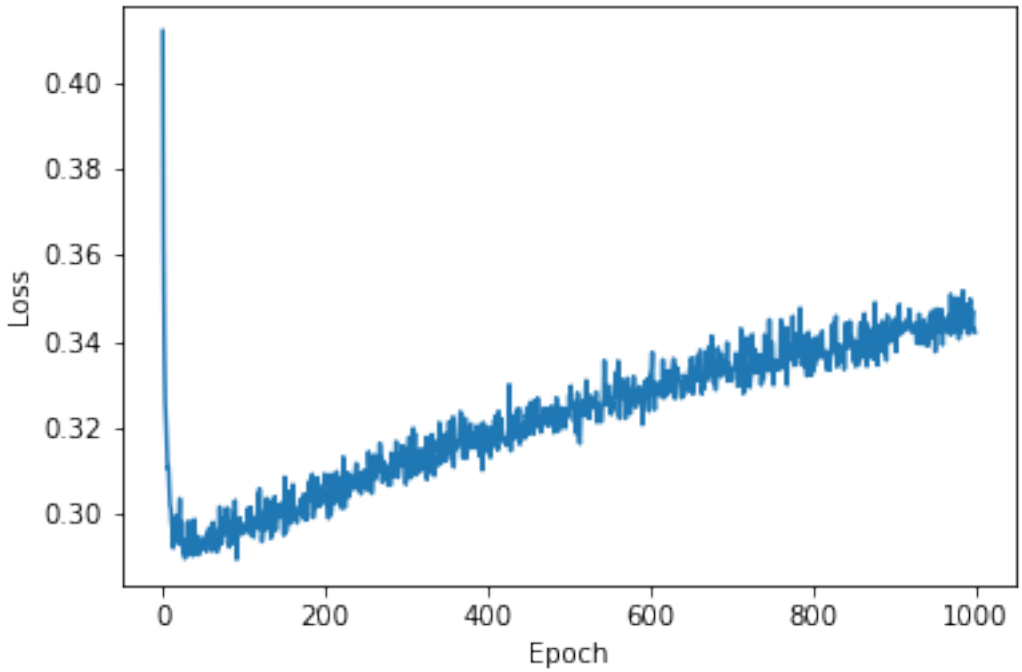

Figure 30. Curve of the NN's loss values over 1000 epochs (validation set)

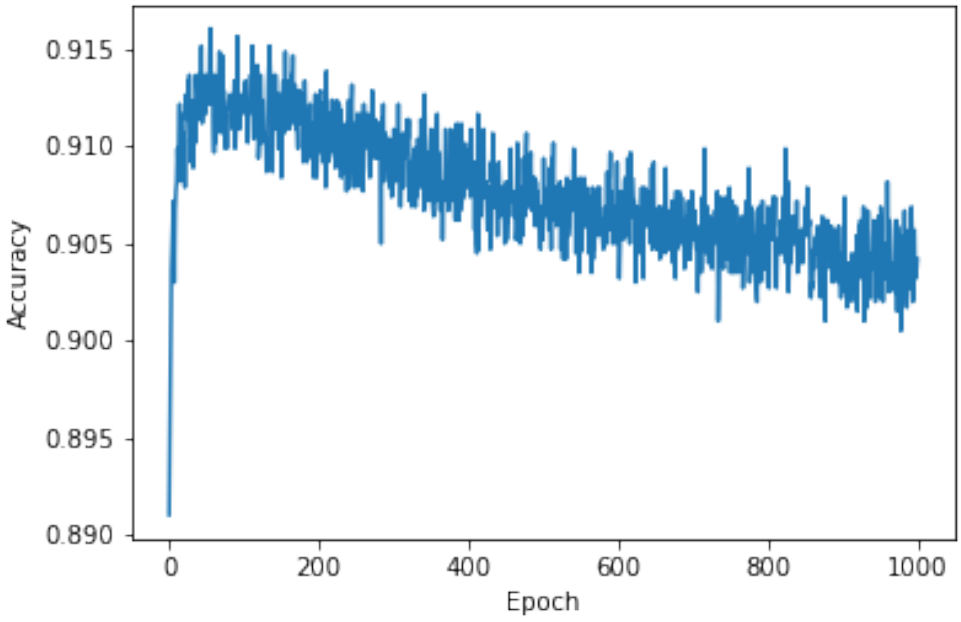

Figure 31. Curve of the NN's accuracy values over 1000 epochs (validation set)

Indeed, as it can be seen from both Figure 30 and Figure 31, the validation curves suggest that with the aforementioned hyperparameter setting, the network requires a small amount of epochs (ca. < 100) to generalize well on unseen data. However, this is not an issue as relevant and as puzzling as it may be thought. It would be if this were the case of the previous phase, but the task of the validation phase is not that of learning or training the NN. Hence, the NN will select exclusively the best combination and values of the hyperparameters, so that the model will perform at its best in the testing phase: this happens whether the values of loss decrease or increase overtime (and same can be affirmed for accuracy).

Thus, considering the two figures just inserted, the best model saved was the one 'represented' at the peak of the curve in Figure 31, which corresponds to low loss values in Figure 30. This model was subsequently saved as a .pth file and loaded in the testing phase.

Finally, coming down to the last phase of the image classification task, the testing phase genuinely assessed the skill and ability of the NN at classifying the images inserted in the test set – which, to remind, had never been seen by the NN. The model was loaded, and it had to simply carry out an image classification task once again and the accuracy was printed at the end of the test phase. The model reached an accuracy of 0.82 on the test set. It is a relatively high and good results, considering the small sample in both the validation and testing sets, but more generally the whole dataset.

For the testing phase, however, this was not the only analysis conducted. The value of the accuracy refers to the general and average accuracy of the entire model; nevertheless, there were 30 classes in total, and it was interesting to analyse the accuracies for each class, to test whether the model under- or overperformed in some more than others. In order to achieve this, a confusion matrix was created, and a list with the accuracies corresponding to each class was printed out. The results are shown in Table 8 below next to their corresponding class[7].

[7] During the NN training, however, these classes were transformed into numbers (from 0 to 29); however, those in the table are the classes taken from the dataset of metaphors, therefore they are essentially nouns.)

Class	Accuracy	Class	Accuracy	Class	Accuracy
Machine	0.9188	Acrobat	0.8873	Rock	0.6875
Flower	0.9482	Person	0.6640	Butterfly	0.9151
Money	0.9500	Lawn	0.6750	Dragonfly	0.7783
Blanket	0.8214	Home	0.8387	Snow	0.6970
Lion	0.9778	Heart	0.5397	Dancer	0.8412
Rug	0.8528	Giraffe	0.9449	Prison	0.8817
Professor	0.8571	Snake	0.7308	Street	0.8412
Swan	0.9394	Sewer	0.8511	Skyscraper	0.8654
Soldier	0.7972	Cloud	0.9504	Cotton	0.7383
Chest	0.8793	Time	0.8272	Armor	0.8672

Table 8. List of classes of the image classification task and their corresponding accuracies in the testing phase

As it can be seen from Table 8, the model performed overall properly, with an average accuracy of above 0.7 for each class, aside from five – namely, *person*, *lawn*, *rock*, *heart*, and *snow*. Four of them have a slightly lower accuracy, between 0.66 and 0.69, which can be anyway considered appropriate. The class with the lowest value in accuracy is *heart*, with ca. 0.54 of accuracy: the reason for this being the case is rather difficult to find; it may be due to the type of images or the number of images contained in the given class.

Notwithstanding this, for the vast majority of the classes, the saved model performed with high results and therefore can be considered successful as far as the image task recognition is concerned.

To sum up, the task – albeit embryonical, given the small dataset available – shows how this proof of concept, together with this type of NN (ResNet50), could be potentially used in a pipeline for visual metaphor interpretation. There are, however, limitations to this task, which are going to be delineated in a few chapters (please refer to 3.7 General discussion).

3.6 The pipeline

As it was explained in 3.4 Experiment 2, the scope of the project was that of creating a way for the machine to interpret visual metaphors. The task involves two sides: recognising the domains in the visual metaphor and use the classes to return adjectives appropriate to describe the metaphor.

As far as the first task is concerned, the processes are more complicated compared to the adjective side. Hence, since there are different steps involved, a more complete pipeline was developed in order to make the system neater. It would also be appropriate to underline that given a certain visual metaphor, the first step would be that of recognising the source and the target contained in the image – this problem has been already discussed in chapter 3.1 The background. This task was, however, undoable, and therefore the pipeline starts from step two.

Once the model has recognised the two visual representations present in the image (one for the source and one for the target), the goal is that of performing a simple classification task – as the one described a few pages before. This process is necessary for the NLP side of the pipeline, since it is based on adjectives related to a certain PoS (namely, the source). Therefore, the classification needs to return the hopefully correct class for the given images.

At this point, a condition has to be inserted: if the class returned by the classification task corresponds 'perfectly' to the noun representing the source or the target in the linguistic conceptual metaphor, a simple printing of that class is sufficient; however, if the class does not match the source or target, it has to be substituted with the more specific class. To make an example, as previously stated, for both 'girl' and 'man' the tar corresponding to the synset 'person' was used (since they both are human beings). It would not be a relevant issue if 'person' is left as a class – in other words, the meaning of the metaphor would not change for humans – but it would have an influence on the type of adjectives retrieved from the chosen source (in this case, Sketch Engine). Hence, the condition would be the following: if the target(s) are *x, y*, and *z*, the class will be substituted with 'girl', otherwise it will be substituted with 'man'. In this way we would allow the model to choose more specifically related adjectives. Finally, the correct classes are printed.

The third step starts the NLP side. Firstly, the model takes as input the classes, and retrieves the adjectives for the source from a chosen database or website. Secondly, it filters them with the *descriptive_adj* set and stores them in a dictionary, which will be used later by the *calc_sim* function. Finally, it calls the function – this can be done in an iteration if multiple visual

metaphors need to be analysed – and this will return the first ten adjectives with the highest similarity.

At this point the only issue would be that of choosing the most appropriate adjective, given the fact that there appears to be no pattern on in the indexes. However, this problem will be addressed in the next section. Below is a pictorial representation of the above-described pipeline.

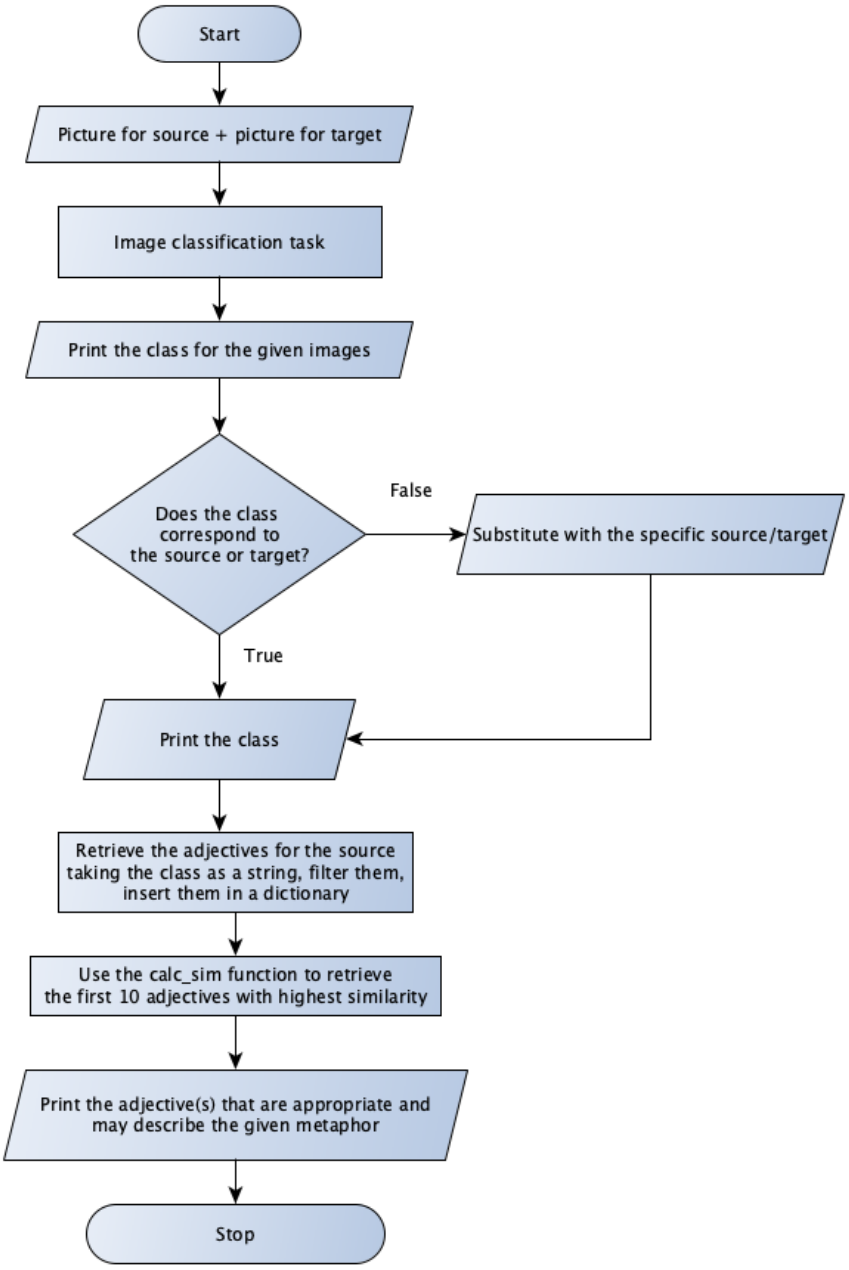

Figure 32. Visual representation of the pipeline for visual metaphor interpretation

Given the fact that both the accuracy for each class in the image classification task and the function's precision at each metaphorical pair are available, it would be sensitive to combine the two analyses in order to obtain a plausible and preliminary accuracy of the model as a whole (for the time being considering only the 20 source-target pairs used for the image classification task). In Table 9 the metaphorical pairs and the corresponding total accuracy are presented[8].

Metaphorical pair	NLP P@10	CV target acc	CV source acc	Overall acc
Acrobat – Butterfly	0.5	0.89	0.91	0.77
Ballerina – Dragonfly	0.4	0.84	0.78	0.67
Giraffe – Skyscraper	0.6	0.94	0.86	0.8
Girl – Armor	0.7	0.66	0.86	0.74
Girl – Flower	0.7	0.66	0.94	0.77
Man – Chest	0.7	0.66	0.87	0.74
Man – Sewer	0.6	0.66	0.85	0.7
Professor – Machine	0.5	0.85	0.92	0.76
Soldier – Lion	0.4	0.8	0.98	0.73
Street – Snake	0.6	0.64	0.73	0.66
Teacher – Rock	0.3	0.85	0.67	0.61
Flower – Blanket	0.7	0.94	0.82	0.82
Snow – Blanket	0.6	0.67	0.82	0.7
Man – Lion	0.6	0.66	0.98	0.75
Ballerina – Swan	0.6	0.84	0.94	0.79
Cloud – Cotton	0.5	0.95	0.74	0.73
Lawn – Rug	0.7	0.67	0.85	0.74
Home – Prison	0.4	0.84	0.88	0.71
Heart – Rock	0.3	0.53	0.67	0.5
Time - Money	0.6	0.82	0.95	0.79

Table 9. Table of overall accuracy of final model per class. NLP P@10 = precision-at-10 for the NLP side of the task. CV target acc = image classification task's accuracy for that target. CV source acc = image classification task's accuracy for that source. Overall accuracy = plausible accuracy for the given couple considering the single accuracies.

[8] The accuracies for the image classification task have been rounded to two decimal numbers.

To conclude, the entire approach, as described by the pipeline in 3.6 The pipeline, plausibly returns overall accuracies above 0.6, which is in itself a robust result, although many of them are above 0.7 or even 0.8. It is possible to find only one instance where the accuracy was below 0.6: for the metaphorical pair *heart-rock*, the overall accuracy plateaus at 0.5, due to both the accuracy for the class *heart* and the precision-at-10 for the NLP side of the pipeline. It is, however, important to underline that this is the case of exclusively one pair out of 20.
Hence, it is safe to assert that the approach and model described by the pipeline are reliable and may be implemented in future work or bigger models. Nevertheless, the approach is not immune to limitations, which will be described in the next section.

3.7 General discussion

The aim of this project was that of finding an adaptable and computationally cheap approach for (visual) metaphor interpretation. As previously depicted, two experiments were carried out, the second being a consequence of the inconclusiveness of the first. There are a few points and topics to discuss; therefore, this chapter plans to go through each of them step by step, in the clearest and most exhaustive way possible.

Experiment 1 consisted in a rather weak method towards a fast and easy metaphor interpretation. Although the expectations for its results were not high, were the experiment to work, it would have been a substantial step forward.

One particular consideration should be carried out regarding the adjectives chosen for the pair *concept – maze*, whose lists' intersection returned some applicable adjectives. Although all these pertinent adjectives are potentially appropriate to be fed to a model in metaphor interpretation and could be used for whatever task is being tackled, it is not enough of a reason to consider this experiment successful. As a matter of fact, it is not possible to define a pattern, which could determine why for certain metaphorical pairs not a single adjective was found, or among the appropriate lists which of the adjectives should be considered (e.g., its index, its position in the lists and so on). Plus, even if the computer does find some appropriate adjectives for a given couple, in the same list there are adjectives that are antonymous. For instance, considering Table 2, it is true that it would be possible to consider 'difficult' as an appropriate adjective; however, the intersection also returned adjectives such as 'simple' or 'easy'. There is not a univocal answer, or multiple answers with related meanings. Here, antonyms are inserted in the same list, from which it should be possible to retrieve at least one possible adjective. Hence, this issue alone is sufficient to say that this method is not solid and should not be proceeded, nor it needs further research.

What is more, even if further research is conducted and a way that would solve the issue is to be found, the problem of novel metaphors would still persist.

Because novel metaphors, as already established, are nowhere to be present in corpora, which need to be used for this experiment, it would be extremely unlikely that the lists of the source and target in an unseen combination will have some appropriate adjectives in common. Experiment 1 as a whole was solely based on the fact that conventional metaphors are highly pervasive. By taking this feature away, the experiment would lose its own structure.

Therefore, it would be nonsensical to keep going down this path, which will lead nowhere, not only as far as novel metaphors are concerned, but also when it comes to conventional metaphors in the first place.

To sum up, the first experiment was to be considered inconclusive and, in addition, did not lead to any worthy opportunity. The idea at the bottom of it was intriguing since it was quite simple and straightforward. If the results were more encouraging, it would have been worth further investigating and potentially insert the approach in a pipeline, since it is not time-consuming at all. However, this was not the case, and that is the reason why a second experiment was conducted.

Experiment 2 was meant to be a valid and anyway fast approach, albeit somewhat more complex. Overall, the analysis returned good results: arguably the most striking feature is the fact that participants always chose at least one adjective per source-target pair; hence, among the first ten adjectives, at least one was considered appropriate to describe the given metaphor. However, this is not to be considered 100% reliable, since – out of experience – participants tend to always choose one of the already given answers, even if it does not fit perfectly as an answer. This is the reason why, firstly, another fill-the-blank choice was made available and, secondly, why a further and more reliable analysis was carried out.

Another interesting result derived from the analysis of precision-at-k, where the highest precision was returned at $k = 1$. It could be argued that since precision-at-1 returned the best accuracy, the model could accept the first adjective return and still interpret the meaning of a (visual) metaphor 72.6% of the times. Nevertheless, for certain metaphors although many participants chose adjective with index 1, a larger number of participants chose another adjective with a different index. Thus, it would be fair to say that this last adjective is 'more appropriate' then the adjective #1 – according to the number of votes. This discrepancy led to the idea that although precision-at-1 has the highest value, precision-at-10 would be more informative. Moreover, considering the fact that there were two thresholds, although the first analysis considered as acceptable even adjectives with only one vote, it does not mean that this analysis is the worse one – and therefore, only the second analysis should be considered.

Indeed, even if only one participant voted for one adjective, it must be considered, because for that person the metaphor could be interpreted in that way. It has been many times established that metaphors are subjective; hence, it would by hypocritical of eliminating adjectives with one vote. Accordingly, precision-at-10 with threshold set at 1 depicts the most explanatory performance of the approach.

The second compelling and surprising result was related to innovativeness. As delineated in the results section of experiment 2, the feature of *innovativeness* returned a p-value of 0.018 in the multilinear regression analysis. The analysis is suggesting that the more innovative a metaphor is, the more likely it is that the machine correctly predicts more adjectives, which could describe it. This is surprising because the expectation involved the exact opposite, namely that the more familiar or comprehensive a certain metaphor was, the higher would have been the precision of the function: indeed, innovative metaphors are less represented in corpora, upon which vector spaces are built. Thus, if the reasoning were correct, it would mean that more familiar metaphors are more frequent, and since they are more frequent in corpora as well, this would lead the function to return better results, because the metaphor has appeared many times. The fact that higher results are linked with higher degree of innovativeness does appear to be counter-intuitive; it is, however, valuable to highlight that this result may have some positive consequences for future work involving novel metaphors. As a matter of fact, it would be reasonable to expect that the function would return more appropriate adjectives if it encountered novel metaphors. The reason for this being the case is the fact that novel metaphors have the highest degree of innovativeness since they have never been uttered before. Therefore, with the idea of generally applying the method to metaphors in mind, this is a huge point in its favour.

As far as the image classification task is concerned, the results can be claimed to be in line with the results in CV, given the small number of images available, both for the training and the testing phases. Indeed, it is fair to assert that the classification of images in visual metaphor interpretation is not the main obstacle, since huge steps have been made in the recent times. The most relevant impediment would involve recognising sources or targets which are not physically present in the image: this may lead to some substantial mistakes. However, human programmers may intervene by making the model guess the absent source or target from other pieces of information in the picture – e.g., a logo, a name, another object which may be related, a text anchor, etc.

The second problem, which was an issue for this project, and which should be resolved, deals with the absence of a robust, rich, and reliable dataset of visual metaphors. It may also be argued that not many visual metaphors are published, and therefore, this may cause the collection to be time-consuming and inconclusive. Unfortunately, in order to obtain good results with image recognition (especially when the bigger pictures contain more than one image representing different subjects) it is necessary to have large datasets. However, with time and patience it may also be possible to overcome this major issue.

A few words should be spent discussing the pipeline, and the possible issues that may arise with it. The problem about image recognition has already been laid out. There may be another concern regarding the choice of adjectives to print out. Since it has been established that precision-at-10 is the most informative, it is also acceptable to claim that it supposedly possible to find an adjective that is appropriate for the metaphor in that range (namely, 10). The matter, however, is that the machine cannot return 10 adjectives, and the reason is twofold. On one hand, it would be redundant to print ten adjectives, so that speakers can choose one; although this would be more 'personal' it is neither effective nor practical. On the other, among these 10 adjectives, there are some which are not appropriate to describe a given metaphor; hence, including them would be a wrong decision.

The question, at this point, would be: which adjective(s) among the 10 should be printed out? The answer is more complex than it actually seems. It would be better to start from the standpoint that a bigger dataset of conceptual metaphors is needed in order to furtherly test the approach. Therefore, it could be possible to use the information retrieved from this work and the future projects, so that machines will already have some baselines. Therefore, simply, the knowledge is stored in the computer and can be recalled when it is necessary.

When it comes to metaphors that have not been processed and stored in previous works, the situation is more complicated. It has been highlighted how *innovativeness* may play an extremely important role: this feature could be used at one's advantage. If after a careful analysis, the precision-at-1 is high enough for novel metaphors, the first adjective in the returned list may be used as descriptor of the metaphor. If this is not the case, a more detailed and carefully thought-through theory should be developed and implemented to test the accuracy.

4. CONCLUSION

The project dealt with finding a possible pipeline for the interpretation of visual metaphors, hence involving both an image classification task and a function manipulating vectors to return adjectives, which could potentially interpret said metaphors.
The results showed how the method is in itself reliable, returning good results especially when the threshold for the precision-at-k analysis was set at 1 (meaning, all adjectives with at least one vote were assigned value 1, the others were assigned value 0). A multilinear regression also demonstrated how more innovative conceptual metaphors were associated with higher results in the function's precision in returning appropriate adjectives. This could lead to significant improvement when testing machines on their performance in interpreting novel metaphors, since novel metaphors are the most innovative metaphors existing.
Moreover, the image classification task confirmed the possibility of using a NN like ResNet50 to correctly classify the images for sources and targets in the conceptual metaphor represented in the visual metaphor. The results were in line with previous work on image classification, given the small dataset, and even the accuracies of each class were overall more than favourable.
Finally, the pipeline appeared to return trustworthy and stable results, by unifying the two sides of the project, and therefore bring the task to a step forward, even if the pipeline proposed technically misses the first step involving the recognition of domains, which would require a further training and testing on a reliable and rich dataset.

4.1 Limitations

Although the results for both the NLP task and the CV task are to be considered significantly good, it would be unfair to state that there are no limitations to this approach. As it will be explained in a few lines, the limitations involve both sides of the project.

As far as the NLP side of the project is concerned, the first limitation for this approach would be where to retrieve the adjectives related to the source, which are fundamental for the function. In this case – i.e., for this project – the choice fell on Sketch Engine (Kilgarriff, Rychlý, Smrz, & Tugwell, 2004), because of the possibility of chosen one particular corpus and have all the functions related to it in one single website, and because of the extremely useful and fast Word

Sketch: for this project the corpus used was the ukWaC (Ferraresi, Zanchetta, Bernardini, & Baroni, 2008); what is more, the words returned through Word Sketch are based on co-occurrences within the corpus chosen, and this facilitates the process. Nonetheless, the ukWaC corpus is not the only source of adjectives (and more in general PoS) available: bigger or more detailed corpora may return different adjectives, or, better, a different list of adjectives. While this may be a minor limitation, since the website used would be always the same, it is a minor limitation that needs, however, to be taken into consideration.

A second possible limitation to this approach may involve the use of fastText (Bojanowski, Grave, Joulin, & Mikolov, 2016). fastText is a static distributional model, and this may limit the way words are represented with vectors. To make an example by taking the word 'house': in fastText, the word 'house' is assigned always the same vector, whether it is inserted in a) or b) (I reinsert the sentences used previously, so it is not necessary to go back to the specific chapter).

c) I can't believe my parents just sold their **house** of 30 years.
d) This **house** was built in 1934.

There are, however, models like BERT (Delvin, Chang, Lee, & Toutanova, 2019) or ELMo (Peters, et al., 2018), where this is not the case, and because a) and b) are two different instances of the word 'house', this will be assigned two different vectors. In fastText, there is the possibility of retrieving and using vectors for ambiguous adjectives, which as a consequence may be inserted in the list, but should not be there, since the ambiguity makes the adjective unsuitable.

Hence, in BERT, because there would be two different vectors, this may have an impact on the results. There is one point to make in regarding this second possible approach: metaphors do not change based on the context. If a speaker says *I'm wasting my time*, while in a certain situation, and then uses the same phrase in a completely different context, *I'm wasting my time* does not change its meaning. The adjectives that could describe the conceptual metaphor TIME IS MONEY, remain the same, in both contexts. Thus, using models like BERT may (or may not) be counter-productive, and a further testing is necessary to define whether this is true or not.

One final limitation, which is worth to lay out, regards the dataset. Indeed, 84 source-target pair is not a small dataset for metaphors per se; yet it is still limited, and this may influence the accuracy of the approach. Thus, increasing the total number of pairs may increase the overall

precision-at-10. However, a discussion should be started since the types of metaphors influences the overall accuracy. As it has been pointed out earlier, the more innovative the metaphor was, the higher the precision.

As far as the Computer Vision side is concerned, it is also possible to find some limitations, which mainly derive from the task of treating metaphors, rather than the classification task itself.

First of all, the most obvious limitation for this particular project concerns the number of images used to represent the sources and targets: since many sources and targets were abstract and since the database for images used was ImageNet (Deng, et al., 2009), many domains were not represented and there was no way of collecting thousands of images for the remaining domains. It is nonetheless an issue, which definitely has an impact on the results of the image classification task, even though it can be affirmed that the results are overall good for such a small dataset. Therefore, increasing the dataset would solve this minor limitation, albeit this will be discussed further in the next section.

Finally, the major problem regarding the CV part of the project is the processing of complete visual metaphors – in other words, simply real and advertised images which have the goal of expressing a metaphor. Indeed, this significant limitation forced the work to be reshaped and redefined, since otherwise it would have been undoable. In terms of a future implementation of this approach, it could be a relevant impediment, which needs to be solved. The mechanism to solve it, yet, may take more than it is expected, since CV is making huge steps in image recognition, but the absence of images may cause real issues to the NN.

4.2 Future works

Given the fact that the results for both the NLP and the CV side are to be considered good, it may be worth spending a few words about the future work which may be conducted on the topic.

As it was delineated in the previous section, one of the issues was the number of source-target pairs collected; this extends to both sides of the project. Already this detail about the dataset may lead to further and perhaps better results and may cast some light over certain patterns, e.g., the link between precision and *innovativeness*. Hence, it would be advisable to spend a few months building a big and rich metaphor of source-target pair and conduct the same psycholinguistic task that Coli (2016) presented in her thesis. This would lead to an increase in the number of images as well, which is vital to increase the accuracy of the NN in recognising the domains. This may however encounter some obstacles: while it is fairly easy to find conceptually metaphorical pairs for language tasks, it is not easy to find visual representations of highly abstract domains. Let us take for instance a very frequent target: 'love'. Every speaker has used 'love' as a target while building an (unconscious) realisation of a conceptual metaphor (e.g., LOVE IS A JOURNEY). If speakers are asked to represent love with a drawing or a picture, they would most likely draw a stylized heart or two people kissing. This is a correct iconography for humans, which already know what love is, they feel it, and they themselves have represented love in that way before. For a NN this is extremely confusing: a stylized heart is always a heart, and it would be problematic for it to distinguish when the stylized heart represents love and when it is simply a stylized heart. It could potentially lead to terrible mistakes, which would have a cascading effect on the NLP side of the pipeline, and therefore on the whole interpretation of the visual metaphor. To sum up, finding images to increase the dataset of images representing the sources and targets may hinder the collection of metaphors.

One further step towards the interpretation of visual metaphors links to the recognition of domains in the bigger image. It is probably safe to say that for the majority of visual metaphors, even if one of the two domains is absent (cf. 1.1.4 Visual metaphors), there are many other features in the images that may help the NN recognise – or predict at least – the left-out domain: e.g., text anchors, brand names, logos, other figures, etc. There may be, however, complex visual metaphors, with which the NN may have significant and perhaps consistent difficulties, and this issue inserted in a bigger model or in context leads to inconclusiveness or a mistake because the NN cannot interpret the metaphor. Hence, it is important that research continues

with respect to image classification, so that it will be possible to find a computationally cheap way to interpret all 'levels' of visual metaphors (from simple and flashy, to hidden and difficult).

Another possible future work which derives from the results given by precision and *innovativeness* would be that of testing the approach (perhaps only the NLP side would be sufficient) on novel metaphors. The first step would be that of creating novel metaphors: since they are novel, they exist nowhere, and creating a dataset from nothing is time-consuming. A solution to this may be that of choosing a list of nouns which normally do not belong to conventional conceptual metaphors and create a survey where participants are asked to create metaphors with the list of nouns given (of the type X IS Y). Once the answers are registered, they should be filtered so as to be sure that there are no conventional metaphors. Finally, the last step would be implementing the function and analyse the results. If the prediction made on this thesis about novel metaphor – and based on the link between precision and *innovativeness* – is correct, the precision-at-10 is expected to be higher than the precision-at-10 with threshold at 1 of experiment 2. On the other hand, if the prediction is false, the precision will be lower. Testing exclusively the NLP side with this particular dataset is not counter-productive since the problem of novel metaphors relies in the language not the visual representation of the domains. Even if the visual metaphor is novel, the NN would still treat it as an image classification task, and therefore the results would, most likely, be the same. However, if it is necessary to further train the NN for other purposes, the whole pipeline can be implemented.

APPENDIX 1

Below is inserted the whole 74-metaphor-long dataset, taken from Coli (2016), together with the psycholinguistic measures that she reported in her thesis.

word_pair	familiarity	quality	innovativeness	valence	concreteness	comprehensibility
acrobata-farfalla	3.1	3.72	3.64	2.27	2.22	5.66
alcol-flagello	5.12	4.56	2.56	-2.02	3.12	5.67
rabbia-veleno	3.45	4.04	3.25	-2.23	5.43	4.34
rabbia-vulcano	5.68	3.56	1.92	-2.11	5.61	6.26
ballerina-libellula	6.49	6.21	3.22	1.58	1.78	5.22
banchieri-avvoltoi	6.22	4.82	5.22	-1.98	2.03	6.01
campana-grido	1.98	4.15	5.78	0.11	2.48	3.2
libro-viaggio	5.98	3.46	3.45	2.56	2.66	5.83
cammello-taxi	5.53	4.83	2.03	0.62	1.22	5.12
cancro-prigione	6.01	5.22	2.65	-2.46	3.45	4.12
bambina-bocciolo	5.98	3.04	4.5	2.4	1.54	4.6
concetto-labirinto	5.45	6.09	5.78	1.24	6.78	5.43
scrupoli-siepe	2.34	3.87	5.45	0.11	6.88	3.28
discorso-cometa	2.18	2.56	6.18	0.89	5.22	3.88
divorzio-voragine	2.89	4.78	3.49	-1.76	3.56	4.79
divorzio-bufera	3.45	5.89	2.56	-1.87	3.48	5.09
occhi-finestre	6.23	5.65	4.34	1.34	1.76	5.89
sentimento-labirinto	2.12	4.87	6.02	2.43	5.13	4.98
amicizia-ancora	4.98	5.48	3.36	2.43	5.76	6.12
amicizia-coperta	2.21	3.56	1.56	2.4	6.02	6.23

giraffa-grattacielo	2.45	2.54	1.98	0.42	1.33	6.12
ragazza-cozza	1.43	3.47	1.56	-1.24	1.76	6.48
ragazza-fiore	1.56	6.24	1.47	2.11	1.88	6.56
pettegolezzo-voragine	3.01	5.67	5.32	-1.78	3.2	5.22
pettegolezzo-virus	5.43	6.26	4.56	-1.09	3.44	3.56
nonno-roccia	6.12	6.03	1.76	2.11	1.89	6.09
idea-prigione	1.29	5.67	4.45	-1.22	4.67	5.43
idee-diamanti	2.32	3.45	5.34	2.12	4.88	4.62
ignoranza-mostro	5.67	3.56	4.59	2.34	5.92	3.98
malattia-tempesta	2.67	4.28	4.78	-1.32	4.32	5.44
insulti-rasoi	3.23	3.22	5.23	-1.55	3.04	3.47
carcere-deliro	3.46	3.92	4.92	-1.98	4.09	4.67
gelosia-mostro	3.23	3.45	2.21	-2.6	5.89	4.04
viaggio-incubo	6.01	6.28	2.98	-1.28	3.01	5.88
cucina-inferno	5.46	5.44	2.36	0.32	2.34	4.84
conoscenza-pozzo	4.43	3.89	2.97	2.56	5.43	4.22
conoscenza-viaggio	4.43	5.83	4.67	2.41	5.21	4.56
avvocato-squalo	6.22	3.92	2.12	0.24	1.86	4.22
lezione-sonnifero	6.01	4.56	1.98	-1.7	2.87	6.09
lettera-bomba	5.32	3.45	3.21	0.54	3.22	5.71
bugia-boomerang	6.42	2.98	5.66	0.22	6.07	3.22
vita-viaggio	6.22	4..45	1.76	2.36	5.66	6.29
amore-croce	5.22	5.43	2.55	-2.2	6.78	4.42
amore-viaggio	5.72	6.12	4.23	2.35	6.32	5.89

pranzo-funerale	2.12	6.12	4.35	-1.45	1.46	3.82
uomo-scrigno	3.48	3.47	3.02	1.87	1.8	4.89
uomo-isola	6.01	4.32	5.87	1.43	2.01	3.94
uomo-fogna	5.92	3.94	2.11	-1.89	1.82	6.29
matrimonio-inferno	1.98	5.12	2.98	-1.31	1.58	4.78
mente-macchina	5.28	6.12	4.98	2.43	2.13	5.82
ministro-trombone	4.53	6.13	2.45	-2.32	1.66	5.89
nipote-ciclone	2.56	4.27	1.98	1.76	2.21	6.24
notizia-bomba	2.12	4.28	3.28	-0.32	2.03	4.82
vecchiaia-tramonto	3.67	5.34	4.56	-1.76	1.78	5.55
opinione-pendolo	4.32	5.22	3.45	-1.03	5.48	4.52
festa-uragano	5.67	5.32	5.67	1.98	2.87	5.67
politico-camaleonte	5.23	4.04	2.12	-2.18	1.78	5.93
professore-pozzo	3.54	4.62	4.32	0.78	2	4.53
professore-macchina	2.26	5.64	2.56	0.98	2.09	4.56
ricercatore-bulldozer	6.01	4.32	5.87	2.43	3.07	3.94
ricercatore-vulcano	3.45	5.91	2.03	2.56	2.78	5.34
senatore-fossile	6.24	5.46	1.93	-1.97	4.58	6.28
ombra-velo	6.54	2.98	6.72	0.34	3.22	3.23
sonno-abbraccio	3.48	3.47	3.02	1.87	1.92	4.89
soldato-leone	1.73	6.23	1.88	1.82	1.57	6.09
strada-serpente	5.78	6.48	1.92	0.22	1.38	6.22
insegnante-roccia	4.53	4.21	2.98	2.32	2.45	3.56
ladri-volpi	4.54	5.12	2.98	-1.98	2.12	5.68
pensiero-uragano	2.35	4.98	3.43	0.87	4.56	5.35

pensiero-iceberg	3.56	4.23	3.94	-1.09	4.52	4.87
pensiero-lancia	3.03	3.12	4.21	-2.19	4.7	4.84
albero-scheletro	3.4	3.13	2.07	-1.4	1.98	3.93
parole-ponti	3.45	4.28	4.73	0.76	2.01	4.11
parole-rasoi	4.34	4.09	4.52	-1.45	2.34	3.84

Table 10. Dataset of 74 metaphorical pair from Coli (2016)

APPENDIX 2

Below the complete results of adjectives returned by the machine (following its order).

	1	2	3	4	5	6	7	8	9	10
Acrobat - butterfly	beautiful	colourful	present	free-flying	endangered	numerous	iridescent	plentiful	likely	common
Alcohol - burden	disproportionate	psychological	unacceptable	substantial	overwhelming	significant	considerable	insurmontable	unreasonable	additional
Anger - venom	particular	effective	powerful	enough	deadly	potent	such	black	toxic	much
Anger - Volcano	spectatular	responsible	impressive	dangerous	explosive	miniature	underwater	gigantic	composite	submerged
Dancer - dragonfly	beautiful	colourful	migrant	golden	adult	large	golden	scarce	giant	beelte
Banker - vulture	corporate	bearded	ravenous	common	yellow	hungry	black	like	adult	rare
Bell - scream	blood curdling	continuous	high-pitched	terrifying	occasional	hysterical	terrible	desperate	horrible	opening

Book - journey	straightforward	extraordinary	interesting	uncomfortable	fascinating	unforgettable	fantastic	worthwhile	spectacular	remarkable
Camel - taxi	purpose-built	long-distance	available	collective	accessible	expensive	dedicated	plentiful	official	ordinary
Cancer - prison	purpose-built	different	notorious	infamous	military	victorian	juvenile	likely	federal	female
Child - bud	underground	discerning	adventitious	darling	little	flowering	young	pointed	healthy	single
Concept - maze	three-dimensional	traditional	mathematical	complicated	fascinating	dimensional	delightful	wonderful	beautiful	labyrinthine
Concern - hedge	important	established	surrounding	attractive	neglected	impenetrable	likely	straight	opposite	right-hand
Discourse - comet	spectatular	brilliant	visible	ancient	periodic	distant	massive	active	famous	short
Divorce - chasm	insurmountable	philosophical	spectacular	ideological	apparent	romantic	frightful	enormous	bottomless	unbridgeable
divorce - storm	revolutionary	catastrophic	devastating	spectacular	horrendous	destructive	tremendous	terrifying	occasional	proverbial

Eye - window	appropriate	interesting	comparable	perpendicular	different	important	possible	sufficient	available	rectangular
Feeling - maze	three-dimensional	wonderful	delightful	complicated	traditional	fascinating	beautiful	difficult	mathematical	confusing
Friendship - anchor	appropriate	emotional	temporary	permanent	powerful	structural	existing	heavy-duty	internal	suitable
Friendship - blanket	personalised	lightweight	protective	available	invisible	colourful	luxurious	handmade	waterproof	electric
Giraffe - skyscraper	impressive	dramatic	massive	downtown	towering	famous	modern	giant	banner	sleek
Girl - armor	beautiful	protective	spiritual	numerical	ancient	whole	black	white	rubber	magic
Girl - flower	beautiful	inconspicuous	spectacular	wonderful	different	attractive	self-pollinated	brilliant	important	available
Gossip - chasm	philosophical	spectacular	insurmontable	ideological	apparent	frightful	bottomless	enormous	romantic	awesome

Gossip - virus	important	responsible	destructive	different	available	widespread	difficult	prevalent	transmissible	dependent
Grandfather - rock	alternative	progressive	present	difficult	similar	classic	weathered	fine-grained	shattered	psychedelic
Idea - prison	different	purpose-built	notorious	infamous	likely	military	victorian	good	juvenile	maximum
Ideas - diamonds	magnificent	brilliant	beautiful	expensive	perfect	industrial	imitation	polycrystalline	little	priceless
Ignorance - monster	uncontrollable	terrible	horrible	terrifying	mythological	undemocratic	mechanical	grotesque	invisible	legendary
Illness - storm	catastrophic	devastating	revolutionary	horrendous	spectacular	destructive	terrible	terrifying	tremendous	occasional
Insults - razors/blades	conventional	straight	disposable	electric	cutthroat	plastic	blunt	sharp	open	cut
prison - frenzy	unprecedented	destructive	collective	speculative	desperate	religious	current	murderous	veritable	initial

Jealousy - monster	uncontrollable	murderous	green-eyed	terrifying	terrible	mythological	horrible	ferocious	legendary	mechanical
Journey - nightmare	technological	terrifying	operational	complete	personal	potential	occasional	apocalyptic	terrible	totalitarian
Kitchen - hell	living	personal	perfect	private	absolute	whole	burning	endless	bloody	fresh
Knowledge - font	interesting	important	possible	available	proportional	preferred	standard	compatible	difficult	specified
Knowledge - journey	straightforward	extraordinary	necessary	interesting	incredible	essential	uncomfortable	fascinating	remarkable	worthwhile
Lawyer - shark	notorious	occasional	resident	mechanical	present	prehistoric	predatory	inflatable	juvenile	budding
Lecture - snooze	comfortable	frequent	little	minute	brief	usual	gentle	restful	quick	light
Letter - bomb	incendiary	thermonuclear	explosive	bouncing	homemade	delayed	capable	close	small	unexploded
Lie - boomerang	cross-shaped	left-handed	massive	autographed	giant	indoor	huge			

Life - journey	straightforward	extraordinary	uncomfortable	interesting	unforgettable	challenging	worthwhile	comfortable	fascinating	incredible
Love - cross	delightful	brilliant	excellent	functional	speculative	attempted	perfect	straight	dangerous	wondrous
Love - journey	straightforward	extraordinary	interesting	wonderful	uncomfortable	fantastic	incredible	unforgettable	fascinating	spectacular
Lunch - funeral	magnificent	alternative	traditional	expensive	splendid	elaborate	republican	ceremonial	dignified	military
Man - chest	mysterious	miniature	exclusive	muscular	victorian	patient	massive	common	painted	hairless
Man - island	spectatular	fascinating	wonderful	beautiful	mysterious	different	attractive	picturesque	important	possible
Man - sewer	experienced	combined	underground	critical	existing	strategic	stinking	defective	victorian	adjacent
Marriage - hell	personal	perfect	absolute	private	eternal	living	whole	burning	endless	bloody
mind-machine	interesting	sophisticated	independent	impossible	necessary	different	acceptable	intelligent	operational	possible
Minister - trombone	principal	second	first	contrabass	fourth	modern	soprano	third	small	tenor

Nephew - tornado	devastating	several	probable	powerful	damaging	violent	common	strong	small	deadly
News - bomb	incendiary	thermonuclear	explosive	bouncing	homemade	delayed	capable	small	close	ready
Old age - sunset	magnificent	breathtaking	sensational	spectacular	fantastic	unforgettable	beautiful	wonderful	incredible	picturesque
Opinion - pendulum	political	unique	compound	electoral	simple	inverted	magnetic	double	mental	heavy
Party - hurricane	catastrophic	devastating	tremendous	destructive	terrible	perfect	powerful	impending	frequnet	third
Politician - chamaeleon	polymorphous	celestial								
Professor - font	interesting	important	proportional	difficult	possible	available	preferred	specified	standard	compatible
Professor - machine	interesting	sophisticated	independent	intelligent	impossible	acceptable	different	excellent	operational	necessary

Researcher - bulldozer	military	armoured	armored	giant	yellow	heavy	huge			
Researcher - volcano	responsible	impressive	spectacular	underwater	dangerous	composite	miniature	explosive	submerged	gigantic
Senator - fossil	interesting	well-preserved	spectacular	fascinating	important	invertebrate	microscopic	original	prehistoric	certain
Shadow - veil	transparent	translucent	impenetrable	diaphanous	universal	corporate	delicate	discreet	light	painted
Sleep - hug	close	then	hard	tight						
Soldier - lion	magnificent	full-grown	wounded	cowardly	ferocious	enormous	sleeping	friendly	female	captive
Street - snake	dangerous	present	non-poisonous	charming	striking	enormous	massive	slippery	poisonous	tropical
Teacher - rock	alternative	progressive	difficult	present	fine-grained	crystalline	metamorphic	psychedelic	similar	suitable
Thief-fox	individual	controlling	responsible	occasional	present	adaptable	resident	little	cunning	likely

Thought - hurricane	terrible	catastrophic	tremendous	devastating	destructive	perfect	fearful	powerful	impending	intense
Thought - iceberg	dangerous	proverbial	enormous	gigantic	floating	submerged	massive	mighty	hidden	dirty
Thought - spear	three-pronged	secondary	ceremonial	ready	bloody	single	flaming	long	short	slender
Tree - skeleton	well-preserved	original	complete	prehistoric	structural	gigantic	similar	cartilaginous	algorithmic	numerous
Word - bridge	necessary	important	picturesque	original	available	essential	difficult	complete	present	dangerous
Word - razor	conventional	straight	disposable	cutthroat	electric	plastic	blunt	sharp	open	cut
flower - blanket	personalised	available	colourful	lightweight	invisible	protective	handmade	luxurious	waterproof	electric
snow - blanket	personalised	available	lightweight	invisible	protective	waterproof	colourful	luxurious	snowy	electric

man - lion	magnificent	full-grown	ferocious	enormous	cowardly	sleeping	friendly	young	literary	famous
dancer - swan	beautiful	resident	graceful	numerous	femal e	majestic	friendly	famous	elegant	proper
cloud - cotton	conventional	comfortable	lightweight	available	important	associate	material	operative	certified	coloured
lawn - carpet	included	complete	cardboard	luxurious	threadbare	synthetic	sumptuous	surface	quality	premier
home - prison	purpose-built	different	notorious	victorian	infamous	military	likely	juvenile	federal	maximum
heart - stone	incorporate	traditional	different	honey-coloured	difficult	original	available	substitute	present	artificial
time - money	beginning	insufficient	necessary	individual	forthcoming	additional	possible	sufficient	important	outstanding
computer - dinosaur	technological	intelligent	complete	terrestrial	prehistoric	political	descendent	warm - blooded	dangerous	hot-blooded

Table 11. Results of experiment 2: adjectives returned by the function calc_sim

Bibliography

Adams, D. (1979). *The Hitchhiker's Guide to the Galaxy.* London, UK: Pen Books.

Barden, J. A., & Lee, M. G. (2002). An artificial intelligence approach to metaphor understanding. *Theoria et Historia Scientiarum, 6*(1), 399-412.

Beigman Klebanov, B., Leong, C., & Flor, M. (2018). A Corpus of Non-Native Written English Annotated for Metaphor. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 86-91). New Orleans, Louisiana: Association for Computational Linguistics.

Bird, S., Edward, L., & Ewan, K. (2009). *Natural Language Processing with Python.* Sebastopol, California: O'Reilly Media In.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *CoRR, abs/1607.04606*. Retrieved January 10, 2022, from fastText: https://fasttext.cc/docs/en/english-vectors.html

Bollegala, D., & Shutova, E. (2013). Metaphor Interpretation Using Paraphrases Extracted from the Web. *PLoS ONE, 8*(9), 1-10.

Bowdle, B. F., & Gentner, D. (2005). The career of metaphor. *Psychological Review, 112*(1), 193–216.

Brownlee, J. (2019, May 1). *A Gentle Introduction to the ImageNet Challenge (ILSVRC).* Retrieved March 2, 2022, from Machine Learning Mastery: https://machinelearningmastery.com/introduction-to-the-imagenet-large-scale-visual-recognition-challenge-ilsvrc/

Chomsky, N. (1957). *Syntactic Structures.* Berlin, Germany: Mouton & Co.

Chouinard, B., Volden, J., Hollinger, J., & Cummine, J. (2019). Spoken metaphor comprehension: Evaluation using the metaphor interference effect. *Discourse Processes, 56*(3), 270-287.

Coli, A. (2016). Quando l'avvocato è uno squalo: studio di categorizzazione di coppie di parole metaforiche e letterali (Master's thesis, University of Pisa, Pisa, Italy). Retrieved from https://etd.adm.unipi.it/t/etd-05302016-122356/

Dankers, V., Rei, M., Lewis, M., & Shutova, E. (2019). Modelling the interplay of metaphor and emotion through multitask learning. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint*

Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 2218–2229). Hong Kong, China: Association for Computational Linguistics.

Deignan, A. (1995). *Collins Cobuild English Guides 7: Metaphor.* London: HarperCollins.

Delvin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 1*, pp. 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F.-F. (2009). ImageNet: A Large-Scale Hierarchical Image Database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248-255). IEEE.

Everingham, M., Gool, L. V., Williams, C., Winn, J., & Zisserman, A. (2012). *PASCAL Visual Object Classes Challenge (VOC)*. Retrieved March 2, 2022, from Pascal Network: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

Fass, D. (1991). met*: A Method for Discriminating Metonymy and Metaphor by Computer. *Computational Linguistics, 17*(1), 49-90.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Cambridge, Massachusetts: MIT Press.

Fellbaum, C., & Miller, G. A. (2003). Morphsemantic links in WordNet. *Traitement automatique de langue, 44*(2), 69-80.

Ferraresi, A., Zanchetta, E., Bernardini, S., & Baroni, M. (2008). Introducing and evaluating ukWaC, a very large Web-derived corpus of English. *Proceedings of the 4th Web as CorpusWorkshop (WAC-4)* (pp. 47-54). Marrakech, Morocco: LREC.

Freud, S. (1914). *Zur Einführung des Narzißmus.* London: Hogarth Press.

Frost, R. (1916). The Road Not Taken. In R. Frost, *Mountain Interval* (p. 9). New York: Henry Holt and Company.

Gao, G., Choi, E., Choi, Y., & Zettlemoyer, L. (2018). Neural metaphor detection in context. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 607-613). Brussels, Belgium: Association for Computational Linguistics.

Geary, J. (2012, October). *I is an Other: The Secret Life of Metaphor and How It Shapes the Way We See the World.* New York: HarperCollins Publisher Inc.

Glucksberg, S., Gildea, P., & Bookin, H. B. (1982). On understanding nonliteral speech: Can people ignore metaphors? *Journal of Verbal Learning and Verbal Behavior, 21*, 85–98.

Glucksberg, S., & Haught, C. (2006). On the relation between metaphor and simile: When comparison fails. *Mind & Language, 21*, 360-278

Gong, H., Gupta, K., Jain, A., & Bhat, S. (2020). IlliniMet: Illinois System for Metaphor Detection with Contextual and Linguistic Information. *Proceedings of the Second Workshop on Figurative Language Processing* (pp. 146–153). Online: Association for Computational Linguistics.

Goodman, J. (2001). Classes for fast maximum entropy training. *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings* (pp. 561-564). IEEE.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning Word Vectors for 157 Languages. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)* (pp. 3483-3487). Miyazaki, Japan: European Language Resources Association (ELRA).

Harris, Z. (1957). Co-occurrence and transformation in linguistic structure. *Language, 33*(3), 283–340.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770-778). IEEE.

Hochreiter, S. (1991). *Untersuchungen zu dynamischen neuronalen Netzen.* Munich, Germany: Technical University Munich, Institute of Computer Science.

ImageNet. (2021, March 11). Retrieved March 9, 2022, from ImageNet: https://www.image-net.org

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2017). Bag of Tricks for Efficient Text Classification. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers* (pp. 427–431). Valencia, Spain: Association for Computational Linguistics.

Kilgarriff, A., Rychlý, P., Smrz, P., & Tugwell, D. (2004). The Sketch Engine. In G. W. Vessier (Ed.), *Proceedings of the 11th EURALEX International Congress* (pp. 105-115). Lorient, France: Université de Bretagne-Sud, Faculté des lettres et des sciences humaines. Retrieved 13 April, from Sketch Engine: https://www.sketchengine.eu

Kilgrarriff, A., Baisa, V., Bušta, J., Jakubíček, M., Kovář, V., Michelfeit, J., . . . Suchomel, V. (2014). The Sketch Engine: ten years on. *Lexicography ASIALEX*(1), 7–36.

Kintsch, W. (2000) Metaphor comprehension: A computational theory. *Psychonomic Bulletin & Review*, *7*, 257–266. https://doi.org/10.3758/BF03212981

Kogan, N., Connor, K., Gross, A., & Fava, D. (1980). Understanding Visual Metaphor: Developmental and Individual Differences. *Monographs of the Society for Research in Child Development, 45*(1), pp. 1-78.

Kovecses, Z., Benczes, R., Bokor, Z., Csabi, S., Lazanyi, O., & Nucz, E. (2010). *Metaphor: A Practical Introduction.* Oxford, England, UK: Oxford University Press.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS 2012)* (pp. 1-9). NIPS.

Lakoff, G. (1994). The Contemporary Theory of Metaphor. In A. Ortony, *Metaphor and Thought (2nd edition).* Cambridge: Cambridge University Press.

Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By.* Chicago: University of Chicago Press.

Lakoff, G., Espenson, J., & Schwartz, A. (1991). *Master Metaphor List.* Berkley: University of California at Berkeley.

Landauer, T. K., & Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, *104*, 211-240

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation, 1*(4), 541-551.

Leong, C., Beigman Klebanov, B., Hamill, C., Stemle, E., Ubale, R., & Chen, X. (2020). A Report on the 2020 VUA and TOEFL Metaphor Detection Shared Task. *Proceedings of the Second Workshop on Figurative Language Processing* (pp. 18-29). Online: Association for Computational Linguistics.

Loper, E., & Bird, S. (2002, July). NLTK: the Natural Language Toolkit. *CoRR, cs.CL/0205028*. Retrieved January 10, 2022, from NLTK: https://www.nltk.org

Manning, C., Raghavan, P., & Schütze, H. (2008). Evaluation in information retrieval. In C. Manning, P. Raghavan, & H. Schütze, *Introduction to Information Retrieval* (pp. 151-175). Cambridge: Cambridge University Press.

Mao, R., Lin, C., & Guerin, F. (2019). End-to-End Sequential Metaphor Identification Inspired by Linguistic Theories. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3888–3898). Florence, Italy: Association for Computational Linguistics.

Martin, J. H. (1990). *A Computational Model of Metaphor Interpretation.* San Diego, CA: Academic Press Professional, Inc.

Mason, Z. J. (2004, March). CorMet: A Computational, Corpus-Based Conventional Metaphor Extraction System. *Computational Linguistics, 30*(1), 24-44.

McGregor, S., Agres, K., Rataj, K., Purver, M., & Wiggins, G. (2019, April 15). Re-Representing Metaphor: Modeling Metaphor Perception Using Dynamically Contextual DIstributional Semantics. *Frontiers in Psychology, 10*(765). Retrieved February 8, 2021, from Frontiers in Psychology.

Metaphor Lab. (2017). *Project*. Retrieved April 9, 2022, from VisMet: http://www.vismet.org/VisMet/project.php

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv*.

Mikolov, T., Grave, E., Bojanowski, P., & Puhrsch, C. J. (2018). Advances in Pre-Training Distributed Word Representations. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).* Miyazaki, Japan: European Language Resources Association (ELRA).

Miller, G. A. (1995, November). WordNet: A lexical Database for English. *Communications of the ACM, 38*(11), 39-41.

Miller, G. A., & Fellbaum, C. (2007). WordNet then and now. *Lang Resources and Evaluation, 41*, 209-214.

Miller, G., Beckwith, R., & Fellbaum, C. (1990). Introduction to WordNet: an on-line Lexical Database. *International Journal of Lexicology, 3*, 235-244. Retrieved February 20, 2022, from Princeton University: https://wordnet.princeton.edu

Mitchell, J., & Lapata, M. (2008). Vector-based models of semantic composition. *Proceedings of ACL* (pp. 236-244). Columbus, OH, USA: Association for Computational Linguistics.

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines . *Proceedings of the 27th International Confer- ence on Machine Learning* (pp. 807–814). Haifa, Israel: Omnipress.

Narayanan, S. (1999). Moving right along: A computational model of metaphoric reasoning about events. *Proceedings of AAAI 99* (pp. 121-128). Orlando, Florida: Association for the Advancement of Artificial Intelligence.

Neidlein, A., Wiesenbach, P., & Markert, K. (2020). An analysis of language models for metaphor recognition. *Proceedings of the 28th International Conference on*

Computational Linguistics (pp. 3722-3736). Online: International Committee on Computational Linguistics.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32* (pp. 8024--8035). Curran Associates, Inc.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (pp. 2227–2237). New Orleans, Louisiana: Association for Computational Linguistics.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Fei-Fei, L. (2015, July 16). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV), 115*(3), 211-252. Retrieved March 2, 2022, from ImageNet: https://www.image-net.org/challenges/beyond_ilsvrc

Rychlý, P. (2007). Manatee/bonito - a modular corpus manager. *Recent Advances in Slavonic Natural Language Processing*, (pp. 65-70). Karlova Studánka, Czech Republic.

Ryoo, Y., Jeon, Y. A., & Sung, Y. (2021). Interpret me! The interplay between visual metaphors and verbal messages in advertising. *International Journal of Advertising, 40*(5), 760-782.

Shakespeare, W. (1623). As You Like It. In W. Shakespeare, *First Folio.* London, England: Edward Blount and William and Isaac Jaggard.

Shutova, E. (2010). Automatic metaphor interpretation as a paraphrasing task. *NAACL* (pp. 1029–1037). Los Angeles, California: Association for Computational Linguistics.

Shutova, E. (2010). Models of Metaphor in NLP. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics* (pp. 688-697). Uppsala: Association for Computational Linguistics.

Shutova, E. (2015). Design and evaluation of metaphor processing systems. *Computational Linguistics, 41*(4), 579-623.

Sidrah. (2021, December 20). *100+ Common Metaphors with Meanings*. Retrieved January 15, 2022, from Leverage Edu: https://leverageedu.com/blog/metaphors/#

Siegler, M. G. (2010, July 20). *Google Image Search: Over 10 Billion Images, 1 Billion Pageviews A Day.* Retrieved February 23, 2022, from TechCrunch: https://techcrunch.com/2010/07/20/google-image-search/?guccounter=1&guce_referrer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&gu

ce_referrer_sig=AQAAAD-9Fw2mwNWXCk59REBmu090CZS2W2svLqderRNsW9lmnicG8lJc_AQ_2VGwCBh1SNSUj4oBGp1A1fJgXImIZ0vuiFdNjpSCA0s6LrS-GSvDtqpf-lFZC-V

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*.

Song, W., Guo, J., Fu, R., Liu, T., & Liu, L. (2021). A Knowledge Graph Embedding Approach for Metaphor Processing. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29*, 406-420.

Steen, G., A.G., D., Herrmann, J., Kaal, A., Krennmayr, T., & Pasma, T. (2010). A method for linguistic metaphor identification. From MIP to MIPVU. Amsterdam, Netherlands.

Stemle, E., & Onysko, A. (2018). Using Language Learner Data for Metaphor Detection. *Proceedings of the Workshop on Figurative Language Processing* (pp. 133-138). New Orleans, Louisiana: Association for Computational Linguistics.

Stowe, K., Moeller, S., Michaelis, L., & Palmer, M. (2019). Linguistic Analysis Improves Neural Metaphor Detection. *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)* (pp. 362–371). Hong Kong, China: Association for Computational Linguistics.

Su, C., Fulumoto, F., Huang, X., Li, J., Wang, R., & Chen, Z. (2020). DeepMet: A Reading Comprehension Paradigm for Token-level Metaphor Detection. *Proceedings of the Second Workshop on Figurative Language Processing* (pp. 30–39). Online: Association for Computational Linguistics.

Su, C., Huang, S., & Chen, Y. (2017). Automatic detection and interpretation of nominal metaphor based on the theory of meaning. *Neurocomputing, 29*, 300-311.

Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going Deeper with Convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1-9). IEEE.

Thibodeau, P., & Boroditsky, L. (2011). Metaphors We Think With: The Role of Metaphor in Reasoning. *PLOS ONE, 6*(2).

Tong, X., Shutova, E., & Lewis, M. (2021). Recent advances in neural metaphor processing: A linguistic, cognitive and social perspective. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 4673-4686). Association for Computational Linguistics.

Treccani. (2021, 01 28). *Emozione. Vocabolario Online*. Retrieved from Treccani: https://www.treccani.it/vocabolario/emozione_res-6b5bee4d-001a-11de-9d89-0016357eee51/

Turney, P., & Pantel, P. (2010). From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research, 37*, 141-188.

Utsumi, A. (2007). Interpretative diversity explains metaphor-simile distinction. *Metaphor and Symbol, 22*, 291-312

Utsumi, A. (2011). Computational exploration of metaphor comprehension processes using a semantic space model. *Cognitive Science 35*, 251–296. doi: 10.1111/j.1551-6709.2010.01144.x

Van Mulken, M., van Hooft, A., & Nederstigt, U. (2014). Finding the tipping point: Visual metaphor and conceptual complexity in advertising. *Journal of Advertising, 43*(4), 333-343.

Vecchi, E. M., Baroni, M., & Zamparelli, R. (2011). (Linear) Maps of the impossible: capturing semantic anomalies in distributional space. *Proceedings of the Workshop on Distributional Semantics and Compositionality* (pp. 1-9). Portland, Oregon, USA: Association for Computational Linguistics.

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Trans. Knowl. Data Eng., 29*(12), 2724-2743.

Word vectors for 157 languages. (2016). Retrieved March 7, 2022, from fastText: https://fasttext.cc

Wu, C., Wu, F., Chen, Y., Wu, S., Yuan, Z., & Huang, Y. (2018). Neural Metaphor Detecting with CNN-LSTM Model. *Proceedings of the Workshop on Figurative Language Processing* (pp. 110-114). New Orleans, Louisiana: Association for Computational Linguistics.

Xu, J., & Du, Q. (2019). A Deep Investigation into FastText. *2019 IEEE 21st International Conference on High Performance Computing and Communications* (pp. 1714-1719). IEEE.

ACKNOWLEDGEMENTS

I would like to spend a few words for those people who have been with me throughout these two difficult, yet satisfying, years and dedicate this thesis to them.
First of all, to my parents, without them I would have not graduated or probably ever gone to university. I love you with all my heart and I enjoyed sharing this journey with you. You were always ready to catch me, were I to fall, and this means the world to me. I owe you everything I have achieved.
To all my relatives, for asking me how my life was going and making me feel appreciated.
To my aunt Cristina, my lucky charm, because let's face it, without your 'good luck' texts, during the exams I would have felt less ready.
To my cousin Giorgia, thanks for taking part in the survey I sent you. I would like to thank you for the laughs and the talking during those few times we saw each other this year as well. I hope those times will double in the future.
To my two best friends, Sharol and Sofia, who took me out of the house, even for a glass of wine: I really needed those nights out. Also, thank you for your constant texts, for the laughs, the suggestions, thank you for existing and being here.

Finally, last but not least I might add, the two most important men of this year (after my dad, obviously).
To Professor Lebani, who joined me in this thesis, without ever judging me or misleading me. It was the greatest pleasure and I worked so well with you. Always kind, spending your precious time talking with me, always ready to give me a hand when I needed it the most. Thank you for the hours spent talking about my future, about the theory to make metaphors work, and for your courses as well. I am sad to see this end, but I also hope we will keep in touch and collaborate again.
And to Dr Torcinovich. You were so busy, but you always found the time to answer to my rather annoying texts. Also, thank you for feeding me to the lions, you taught me 70% of what I know about programming. I hope we are not going separate ways, but if we are, thank you for being you and I wish you all the greatest achievements.